

Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/JP05/004676

International filing date: 16 March 2005 (16.03.2005)

Document type: Certified copy of priority document

Document details: Country/Office: JP
Number: 2004-086174
Filing date: 24 March 2004 (24.03.2004)

Date of receipt at the International Bureau: 28 April 2005 (28.04.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)



World Intellectual Property Organization (WIPO) - Geneva, Switzerland
Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application: 2 0 0 4 年 3 月 2 4 日

出 願 番 号
Application Number: 特 願 2 0 0 4 - 0 8 6 1 7 4

パリ条約による外国への出願
に用いる優先権の主張の基礎
となる出願の国コードと出願
番号

The country code and number
of your priority application,
to be used for filing abroad
under the Paris Convention, is

J P 2 0 0 4 - 0 8 6 1 7 4

出 願 人
Applicant(s): 松下電器産業株式会社

2 0 0 5 年 4 月 1 3 日

特許庁長官
Commissioner,
Japan Patent Office

小 川



【書類名】	特許願
【整理番号】	2037950002
【あて先】	特許庁長官殿
【国際特許分類】	G06F 12/08 310
【発明者】	
【住所又は居所】	大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内
【氏名】	中西 龍太
【発明者】	
【住所又は居所】	大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内
【氏名】	岡林 はづき
【発明者】	
【住所又は居所】	大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内
【氏名】	田中 哲也
【発明者】	
【住所又は居所】	大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内
【氏名】	清原 督三
【特許出願人】	
【識別番号】	000005821
【氏名又は名称】	松下電器産業株式会社
【代理人】	
【識別番号】	100109210
【弁理士】	
【氏名又は名称】	新居 広守
【手数料の表示】	
【予納台帳番号】	049515
【納付金額】	21,000円
【提出物件の目録】	
【物件名】	特許請求の範囲 1
【物件名】	明細書 1
【物件名】	図面 1
【物件名】	要約書 1
【包括委任状番号】	0213583

【書類名】 特許請求の範囲

【請求項 1】

プロセッサの状態に関する条件を生成する条件生成手段と、
現在のプロセッサの状態が前記条件を満たすかどうかを判定する判定手段と、
操作対象となるアドレスを生成するアドレス生成手段と、
前記判定手段が条件を満たすと判定したときに前記アドレス生成手段によって生成されたアドレスを用いてキャッシュを操作する操作手段と
を備えることを特徴とするキャッシュメモリシステム。

【請求項 2】

前記条件生成手段は、前記判定手段が条件を満たすと判定した場合に新たな条件を生成する

ことを特徴とする請求項 1 記載のキャッシュメモリシステム。

【請求項 3】

前記条件生成手段は、プロセッサ内の特定レジスタの値に関する条件を生成する

ことを特徴とする請求項 2 記載のキャッシュメモリシステム。

【請求項 4】

前記特定レジスタはプログラムカウンタである

ことを特徴とする請求項 3 記載のキャッシュメモリシステム。

【請求項 5】

前記条件生成手段は、特定のアドレス範囲内へのメモリアクセスおよび特定のアドレス範囲外へのメモリアクセスの何れかを前記条件として生成する

ことを特徴とする請求項 2 記載のキャッシュメモリシステム。

【請求項 6】

前記条件生成手段は、プロセッサが特定命令を実行することを前記条件として生成する

ことを特徴とする請求項 1 記載のキャッシュメモリシステム。

【請求項 7】

前記条件生成手段は、現在の条件に特定の演算を施すことによって前記新たな条件を生成する

ことを特徴とする請求項 2 記載のキャッシュメモリシステム。

【請求項 8】

前記条件生成手段はメモリアクセスアドレスを条件として生成し、

前記判定手段が条件を満たすと判定した場合に現在の条件に定数を加算することによって前記新たな条件を生成する

ことを特徴とする請求項 7 記載のキャッシュメモリシステム。

【請求項 9】

前記定数は、プロセッサにより実行されるポストインクリメント付きロード／ストア命令におけるインクリメント値またはデクリメント値、およびプロセッサにより実行される 2 回のロード／ストア命令におけるアドレスの差分値の何れかである

ことを特徴とする請求項 8 記載のキャッシュメモリシステム。

【請求項 10】

前記条件生成手段は複数の条件を生成し、

前記判定手段は、複数の条件のすべてを満たすかどうかを判定する

ことを特徴とする請求項 1 記載のキャッシュメモリシステム。

【請求項 11】

前記条件生成手段は複数の条件を生成し、

前記判定手段は、複数の条件の何れかを満たすかどうかを判定する

ことを特徴とする請求項 1 記載のキャッシュメモリシステム。

【請求項 12】

前記操作手段は、

前記判定手段が条件を満たすと判定したときに、前記アドレス生成手段により生成され

たアドレスに対応するデータがキャッシュに格納されているかどうかを判定するデータ判定手段と、

格納されていないと判定された場合に、キャッシュメモリ中のラインを選択する選択手段と、

前記選択されたラインが有効でダーティならライトバックを行うライトバック手段と、

前記アドレスに対応するデータをメモリからライトバック後の選択されたラインへ転送する転送手段と、

前記アドレスをタグとして前記選択されたラインへ登録する登録手段と

を備えることを特徴とする請求項 1 から 3 の何れかに記載のキャッシュメモリシステム

。

【請求項 1 3】

前記操作手段は、

前記判定手段が条件を満たすと判定したときに、前記アドレス生成手段により生成されたアドレスに対応するデータがキャッシュに格納されているかどうかを判定するデータ判定手段と、

格納されていないと判定された場合に、キャッシュメモリ中のラインを選択する選択手段と、

選択されたラインが有効でダーティであれば、ライトバックを行うライトバック手段と、

、

メモリから選択されたラインへデータを転送することなく、前記生成したアドレスをタグとして選択されたラインへ登録する登録手段と

を備えることを特徴とする請求項 1 から 3 の何れかに記載のキャッシュメモリシステム

。

【請求項 1 4】

前記操作手段は、

前記判定手段が条件を満たすと判定したときに、前記アドレス生成手段により生成されたアドレスに対応するデータがキャッシュに格納されているかどうかを判定するデータ判定手段と、

格納されていると判定された場合に、キャッシュメモリ中の格納先のラインを選択する選択手段と、

選択されたラインが有効でかつダーティであればライトバックを行うライトバック手段と、

を備えることを特徴とする請求項 1 から 3 の何れかに記載のキャッシュメモリシステム

。

【請求項 1 5】

前記操作手段は、

前記判定手段が条件を満たすと判定したときに、前記アドレス生成手段により生成されたアドレスに対応するデータがキャッシュに格納されているかどうかを判定するデータ判定手段と、

格納されていると判定された場合に、キャッシュメモリ中の格納先のラインを選択する選択手段と、

選択されたラインを無効化する無効化手段と

を備えることを特徴とする請求項 1 から 3 の何れかに記載のキャッシュメモリシステム

。

【請求項 1 6】

前記操作手段は、

前記判定手段が条件を満たすと判定したときに、前記アドレス生成手段により生成されたアドレスに対応するデータがキャッシュに格納されているかどうかを判定するデータ判定手段と、

格納されていると判定された場合に、キャッシュメモリ中の格納先のラインを選択する

選択手段と、

ラインのアクセス順序を示す順序情報に対して、選択されたラインのアクセス順序を変更する変更手段と、

を備えることを特徴とする請求項 1 から 3 の何れかに記載のキャッシュメモリシステム。

【請求項 1 7】

前記条件生成手段により前記条件としてメモリアドレスを生成し、

前記操作手段は、さらに、

前記条件生成手段により生成されたメモリアドレスがラインの途中を指す場合に、当該ラインの先頭、次のラインの先頭および前のラインの先頭の何れかを指すように調整することによりアドレスを生成する調整手段を備える

ことを特徴とする請求項 1 2 から 1 6 の何れかに記載ののキャッシュシステム。

【請求項 1 8】

キャッシュメモリの制御方法であって、

プロセッサの状態に関する条件を生成する条件生成ステップと、

現在のプロセッサの状態が前記条件を満たすかどうかを判定する判定ステップと、

操作対象となるアドレスを生成するアドレス生成ステップと、

前記判定ステップにおいて条件を満たすと判定したときに前記アドレス生成ステップにおいて生成されたアドレスを用いてキャッシュを操作する操作ステップと

を有することを特徴とする制御方法。

【書類名】 明細書

【発明の名称】 キャッシュメモリ及びその制御方法

【技術分野】

【0001】

本発明は、プロセッサのメモリアクセスを高速化するためのキャッシュメモリ及びその制御方法に関する。

【背景技術】

【0002】

近年のマイクロプロセッサでは、例えば、SRAM (S t a t i c R a n d o m A c c e s s M e m o r y) 等から成る小容量で高速なキャッシュメモリをマイクロプロセッサの内部、もしくはその近傍に配置し、データの一部をキャッシュメモリに記憶することによって、マイクロプロセッサのメモリアクセスを高速化させている。

【0003】

キャッシュの効率向上(ヒット率向上、キャッシュミスレイテンシ低減)のため、キャッシュミスが発生する前に、近い未来に使用するデータを予めキャッシュにフィルする技術としてプリロード(又はプリフェッチ)がある(例えば特許文献1)。

【0004】

従来のプリフェッチ技術では、プリフェッチ命令により指定したアドレスを含むラインをキャッシュにロードしている。これにより、キャッシュミスの低減を図っている。

【特許文献1】 特開平7-295882号公報

【発明の開示】

【発明が解決しようとする課題】

【0005】

しかしながら、上記従来技術によれば、例えば、ソフトウェアによって、ループの外側で一括してプリフェッチを行った場合、ループで必要なデータ領域をあらかじめすべてキャッシュ上に確保することになるため、キャッシュの容量が小さい場合には、それ以外の必要なデータがキャッシュから追い出され、キャッシュミスが発生する。また、データのキャッシング終了、無効化等をループの外側で一括して行う場合は、ループを抜けるまでこれらの動作によるキャッシュの開放が行われないため、キャッシュの容量が不足し、キャッシュミスが発生する。

【0006】

また、ループ中にソフトウェアによってキャッシュ操作を行う命令を挿入した場合、キャッシュ操作を行うアドレスをループの中でソフトウェアによって管理する必要がある。つまり、キャッシュ操作を行う命令をループの中に記述する必要があるため、性能の劣化が発生する。

【0007】

さらに、メモリへのアクセスの状況をハードウェアによって監視し、ハードウェアによって自動的にキャッシュ操作をする場合、正確な予測が行えないと無駄な転送が発生する、あるいは、ソフトウェアからの正確な情報がないとキャッシュ上のデータと外部メモリ上のデータの整合性が取ることができないため、ハードウェアの予測によってこれらの操作を行うことは困難である。

【課題を解決するための手段】

【0008】

上記課題を解決するため本発明のキャッシュメモリシステムは、プロセッサの状態に関する条件を生成する条件生成手段と、現在のプロセッサの状態が前記条件を満たすかどうかを判定する判定手段と、操作対象となるアドレスを生成するアドレス生成手段と、前記判定手段が条件を満たすと判定したときに前記アドレス生成手段によって生成されたアドレスを用いてキャッシュを操作する操作手段とを備える。

【0009】

ここで、前記条件生成手段は、前記判定手段が条件を満たすと判定した場合に新たな条

件を生成するようにしてもよい。

この構成によれば、プロセッサの動作状態が条件を満たすようになったときにキャッシュを操作するので、プロセッサの動作の進行状況に同期してキャッシュメモリを操作することができる。またソフトウェア的に実現しないのでプロセッサに負荷をかけることなくキャッシュメモリを効率よく性能劣化を招くことなく動作させることができる。

【0010】

ここで、前記条件生成手段は、プロセッサ内の特定レジスタの値に関する条件を生成するようにしてもよい。前記特定レジスタはプログラムカウンタであってもよい。

この構成によれば、メモリアクセスアドレスや、プログラムフェッチアドレスを前記条件として、プロセッサの状態を監視することができる。

【0011】

ここで、前記条件生成手段は、特定のアドレス範囲内へのメモリアクセスおよび特定のアドレス範囲外へのメモリアクセスの何れかを前記条件として生成するようにしてもよい。

【0012】

また、前記条件生成手段は、プロセッサが特定命令を実行することを前記条件として生成するようにしてもよい。

ここで、前記条件生成手段は、現在の条件に特定の演算を施すことによって前記新たな条件を生成するようにしてもよい。

【0013】

また、前記条件生成手段はメモリアクセスアドレスを条件として生成し、前記判定手段が条件を満たすと判定した場合に現在の条件に定数を加算することによって前記新たな条件を生成するようにしてもよい。

【0014】

ここで、前記定数は、プロセッサにより実行されるポストインクリメント付きロード／ストア命令におけるインクリメント値またはデクリメント値、およびプロセッサにより実行される2回のロード／ストア命令におけるアドレスの差分値の何れかであるようにしてもよい。

【0015】

ここで、前記条件生成手段は複数の条件を生成し、前記判定手段は、複数の条件のすべてを満たすかどうかを判定するようにしてもよい。

また、前記条件生成手段は複数の条件を生成し、前記判定手段は、複数の条件の何れかを満たすかどうかを判定するようにしてもよい。

【0016】

ここで、前記操作手段は、前記判定手段が条件を満たすと判定したときに、前記アドレス生成手段により生成されたアドレスに対応するデータがキャッシュに格納されているかどうかを判定するデータ判定手段と、格納されていないと判定された場合に、キャッシュメモリ中のラインを選択する選択手段と、前記選択されたラインが有効でダーティならライトバックを行うライトバック手段と、前記アドレスに対応するデータをメモリからライトバック後の選択されたラインへ転送する転送手段と、前記アドレスをタグとして前記選択されたラインへ登録する登録手段とを備えるようこうせいしてもよい。

【0017】

この構成によれば、キャッシュメモリのプリフェッチを、プロセッサの動作状況を監視して同期を取りつつ適切なタイミングで行うことができる。

ここで、前記操作手段は、前記判定手段が条件を満たすと判定したときに、前記アドレス生成手段により生成されたアドレスに対応するデータがキャッシュに格納されているかどうかを判定するデータ判定手段と、格納されていないと判定された場合に、キャッシュメモリ中のラインを選択する選択手段と、選択されたラインが有効でダーティであれば、ライトバックを行うライトバック手段と、メモリから選択されたラインへデータを転送することなく、前記生成したアドレスをタグとして選択されたラインへ登録する登録手段と

を備える構成としてもよい。

【００１８】

この構成によれば、キャッシュメモリのラインにデータを転送するとなく確保すること（タッチと呼ぶ）を、プロセッサの動作状況を監視して同期を取りつつ適切なタイミングで行うことができる。

【００１９】

ここで、前記操作手段は、前記判定手段が条件を満たすと判定したときに、前記アドレス生成手段により生成されたアドレスに対応するデータがキャッシュに格納されているかどうかを判定するデータ判定手段と、格納されていると判定された場合に、キャッシュメモリ中の格納先のラインを選択する選択手段と、選択されたラインが有効でかつダーディであればライトバックを行うライトバック手段とを備える構成としてもよい。

【００２０】

この構成によれば、キャッシュメモリのラインデータのライトバックを（キャッシング終了）を、プロセッサの動作状況を監視して同期を取りつつ適切なタイミングで行うことができる。

【００２１】

ここで、前記操作手段は、前記判定手段が条件を満たすと判定したときに、前記アドレス生成手段により生成されたアドレスに対応するデータがキャッシュに格納されているかどうかを判定するデータ判定手段と、格納されていると判定された場合に、キャッシュメモリ中の格納先のラインを選択する選択手段と、選択されたラインを無効化する無効化手段とを備える構成としてもよい。

【００２２】

この構成によれば、キャッシュメモリのラインの無効化を、プロセッサの動作状況を監視して同期を取りつつ適切なタイミングで行うことができる。

ここで、前記操作手段は、前記判定手段が条件を満たすと判定したときに、前記アドレス生成手段により生成されたアドレスに対応するデータがキャッシュに格納されているかどうかを判定するデータ判定手段と、格納されていると判定された場合に、キャッシュメモリ中の格納先のラインを選択する選択手段と、ラインのアクセス順序を示す順序情報に対して、選択されたラインのアクセス順序を変更する変更手段とを備える構成としてもよい。

【００２３】

この構成によれば、キャッシュメモリのラインのアクセス順序情報を、プロセッサの動作状況を監視して同期を取りつつ適切なタイミングで行うことができる。これによりいわゆるLRUによるキャッシュのリプレース順序を早めたり遅くしたりすることができる。

【００２４】

ここで、前記条件生成手段により前記条件としてメモリアドレスを生成し、前記操作手段は、さらに、前記条件生成手段により生成されたメモリアドレスがラインの途中を指す場合に、当該ラインの先頭、次のラインの先頭および前のラインの先頭の何れかを指すように調整することによりアドレスを生成する調整手段を備える構成としてもよい。

【００２５】

この構成によれば、プロセッサがアドレス昇順にアクセスする場合も、アドレス高順にアクセスする場合も、次に必要なラインのアドレスを適切に算出することができる。

また、本発明のキャッシュメモリの制御方法についても上記と同様の手段、作用を有する。

【発明の効果】

【００２６】

本発明のキャッシュメモリによれば、プロセッサの動作の進行状況に同期してキャッシュメモリを操作することができる。またソフトウェア的に実現しないのでプロセッサに負荷をかけることなくキャッシュメモリを効率よく性能劣化を招くことなく動作させることができる。

【0027】

例えば、プロセッサがポストインクリメント付きロード命令によってシーケンシャルにメモリアクセスする場合には、効率よくプリフェッチすることができる。また、プロセッサがポストインクリメント付きストア命令によってシーケンシャルにデータを書き込む場合には、メモリからキャッシュメモリにデータをロードするペナルティを削除することができ、さらに効率よくキャッシュエントリーをタッチ（確保）することができる。

【0028】

また、プロセッサがポストインクリメント付きストア命令によってシーケンシャルにデータを書き込む場合に、プロセッサのストア命令の進行状況に応じて、ストアが完了したラインを予測し、予測したラインにキャッシング終了属性や無効化属性を設定するので、プロセッサでは、キャッシュメモリのラインサイズやライン境界を認識しなくても、ライトバックすべきラインをキャッシュメモリにおいて予測し、効率よくクリーニング（ライトバック）や、キャッシュエントリーの開放（無効化）を行うことができる。

【0029】

さらに、プロセッサがポストインクリメント付きロード命令によってシーケンシャルにデータを読み出す場合に、読み出しが完了したラインデータを保持するキャッシュエントリーにリプレース属性が設定され、真っ先にリプレース対象として選択されるので、アクセス頻度の低いデータがキャッシュメモリに残ることによるキャッシュミスの誘発を低減することができる。

【発明を実施するための最良の形態】

【0030】

（実施の形態1）

＜全体構成＞

図1は、本発明の実施の形態1におけるプロセッサ1、キャッシュメモリ3、メモリ2を含むシステムの概略構成を示すブロック図である。同図のように、本発明のキャッシュメモリ3は、プロセッサ1およびメモリ2を有するシステムに備えられる。

【0031】

キャッシュメモリ3は、プロセッサ1から指定された条件に従って、予測プリフェッチを行うよう構成されている。予測プリフェッチとは、プロセッサ1によるメモリアクセスの進行状況に基づいて次にプリフェッチすべきラインアドレスを予測し、予測したラインアドレスのデータをキャッシュメモリにプリフェッチすることをいう。また、プロセッサ1から指定される条件というのは、アドレス範囲、シーケンシャルアクセスする場合のアドレスのインクリメント値またはデクリメント値、またはアクセス方向（アドレス昇順または降順）等である。予測プリフェッチは、プロセッサがシーケンシャルにメモリアクセスする場合に適している。

【0032】

加えて、キャッシュメモリ3は、プロセッサ1から指定された条件に従って、予測タッチを行うよう構成されている。予測タッチとは、プロセッサ1によるメモリアクセスの進行状況に基づいて次にタッチすべきラインアドレスを予測し、予測したラインアドレスのデータをロードすることなくセットキャッシュエントリーにバリッドフラグとタグとを設定することにより確保することをいう。予測タッチは、配列データなどの演算結果をメモリに順次格納する場合などに適している。

【0033】

＜キャッシュメモリの構成＞

以下、キャッシュメモリ3の具体例として、4ウェイ・セット・アソシエイティブ方式のキャッシュメモリに本発明を適用した場合の構成について説明する。

【0034】

図2は、キャッシュメモリ3の構成例を示すブロック図である。同図のように、キャッシュメモリ3は、アドレスレジスタ20、メモリI/F21、デコード30、4つのウェイ31a～31d（以下ウェイ0～3と略す）、4つの比較器32a～32d、4つのア

ンド回路 3 3 a ～ 3 3 d、オア回路 3 4、セレクト 3 5、3 6、デマルチプレクサ 3 7、制御部 3 8 を備える。

【 0 0 3 5 】

アドレスレジスタ 2 0 は、メモリ 2 へのアクセスアドレスを保持するレジスタである。このアクセスアドレスは 3 2 ビットであるものとする。同図に示すように、アクセスアドレスは、最上位ビットから順に、2 1 ビットのタグアドレス、4 ビットのセットインデックス（図中の S I）、5 ビットのワードインデックス（図中の W I）を含む。ここで、タグアドレスはウェイにマッピングされるメモリ中の領域（そのサイズはセット数×ブロックである）を指す。この領域のサイズは、タグアドレスよりも下位のアドレスビット（A 1 0 ～ A 0）で定まるサイズつまり 2 k バイトであり、1 つのウェイのサイズでもある。セットインデックス（S I）はウェイ 0 ～ 3 に跨る複数セットの 1 つを指す。このセット数は、セットインデックスが 4 ビットなので 1 6 セットある。タグアドレスおよびセットインデックスで特定されるキャッシュエントリは、リプレース単位であり、キャッシュメモリに格納されている場合はラインデータ又はラインと呼ばれる。ラインデータのサイズは、セットインデックスよりも下位のアドレスビットで定まるサイズつまり 1 2 8 バイトである。1 ワードを 4 バイトとすると、1 ラインデータは 3 2 ワードである。ワードインデックス（W I）は、ラインデータを構成する複数ワード中の 1 ワードを指す。アドレスレジスタ 2 0 中の最下位 2 ビット（A 1、A 0）は、ワードアクセス時には無視される。

【 0 0 3 6 】

メモリ I/F 2 1 は、キャッシュメモリ 3 からメモリ 2 へのデータのライトバックや、メモリ 2 からキャッシュメモリ 3 へのデータのロード等、キャッシュメモリ 3 からメモリ 2 をアクセスするための I/F である。

【 0 0 3 7 】

デコード 3 0 は、セットインデックスの 4 ビットをデコードし、4 つのウェイ 0 ～ 3 に跨る 1 6 セット中の 1 つを選択する。

4 つのウェイ 0 ～ 3 は、同じ構成を有数する 4 つのウェイであり、4 × 2 k バイトの容量を有する。各ウェイは、1 6 個のキャッシュエントリを有する。1 つのキャッシュエントリは、バリッドフラグ V、2 1 ビットのタグ、1 2 8 バイトのラインデータ、ダーティフラグ D を有する。タグは 2 1 ビットのタグアドレスのコピーである。ラインデータは、タグアドレスおよびセットインデックスにより特定されるブロック中の 1 2 8 バイトデータのコピーである。バリッドフラグ V は、当該キャッシュエントリのデータが有効か否かを示す。ダーティフラグ D は、当該キャッシュエントリにプロセッサから書き込みがあったか否か、つまりサブライン中にキャッシュされたデータが存在するが書き込みによりメモリ中のデータと異なるためメモリに書き戻すことが必要か否かを示す。

【 0 0 3 8 】

比較器 3 2 a は、アドレスレジスタ 2 0 中のタグアドレスと、セットインデックスにより選択されたセットに含まれる 4 つのタグ中のウェイ 0 のタグとが一致するか否かを比較する。比較器 3 2 b ～ 3 2 c についても、ウェイ 3 1 b ～ 3 1 d に対応すること以外は同様である。

【 0 0 3 9 】

アンド回路 3 3 a は、バリッドフラグと比較器 3 2 a の比較結果とが一致するか否かを比較する。この比較結果を h 0 とする。比較結果 h 0 が 1 である場合は、アドレスレジスタ 2 0 中のタグアドレスおよびセットインデックスに対応するラインデータが存在すること、つまりウェイ 0 においてヒットしたことを意味する。比較結果 h 0 が 0 である場合は、ミスヒットしたことを意味する。アンド回路 3 3 b ～ 3 3 d についても、ウェイ 3 1 b ～ 3 1 d に対応すること以外は同様である。その比較結果 h 1 ～ h 3 は、ウェイ 1 ～ 3 でヒットしたかミスしたかを意味する。

【 0 0 4 0 】

オア回路 3 4 は、比較結果 h 0 ～ h 3 のオアをとる。このオアの結果を h i t とする。

hitは、キャッシュメモリにヒットしたか否かを示す。

セクタ35は、選択されたセットにおけるウェイ0～3のラインデータのうち、ヒットしたウェイのラインデータを選択する。

【0041】

セクタ36は、セクタ35により選択された32ワードのラインデータのうち、ワードインデックスに示される1ワードを選択する。

デマルチプレクサ37は、キャッシュエントリーにデータを書き込む際に、ウェイ0～3の1つに書き込みデータを出力する。この書き込みデータはワード単位でよい。

【0042】

制御部38は、予測処理部39を含み、キャッシュメモリ3の全体の制御を行う。予測処理部39は、主としてプリフェッチの制御を行う。

【0043】

<予測処理部の構成>

図3は、予測処理部39の構成例を示すブロック図である。同図のように予測処理部39は、コマンドレジスタ401、スタートアドレスレジスタ402、サイズレジスタ403、加算器404、スタートアライナ405a、405b、エンドアライナ406a、406b、アクセスアドレスレジスタ407、予測値レジスタ408、定数レジスタ409、セクタ410、加算器411、比較器412、実行部413を備える。

【0044】

コマンドレジスタ401は、プロセッサ1から直接アクセス可能なレジスタであり、プロセッサ1により書き込まれたコマンドを保持する。図4(c)に、コマンドレジスタ401にコマンドを書き込む命令の一例を示す。この命令は、通常の転送命令(mov命令)であり、ソースオペランドとしてコマンドを、デスティネーションオペランドとしてコマンドレジスタ(CR)401を指定している。図4(d)に、コマンドフォーマットの一例を示す。このコマンドフォーマットは、コマンド内容と定数とからなる。ここで、コマンド内容は、予測プリフェッチコマンドと、予測タッチコマンドとの何れかを表す。定数は、プロセッサ1が、シーケンシャルにメモリアクセスする場合のメモリアクセスアドレスのインクリメント値またはデクリメント値を表し、例えば、+4、+8、-4、-8等である。なお、コマンド中の定数の代わりに、アクセス方向(アドレス昇順かアドレス降順か)と絶対値(連続アクセスする場合のアドレスの差分)とを含むようにしてもよい。

【0045】

スタートアドレスレジスタ402は、プロセッサ1から直接アクセス可能なレジスタであり、プロセッサ1により書き込まれたスタートアドレスを保持する。このスタートアドレスはCフラグ(キャッシングを終了してよいか否かを示すクリーニングフラグ)を設定すべきアドレス範囲の開始位置を示す。図4(a)に、スタートアドレスレジスタ402にスタートアドレスを書き込む命令の一例を示す。この命令も、図4(c)と同様に通常の転送命令(mov命令)である。

【0046】

サイズレジスタ403は、プロセッサ1から直接アクセス可能なレジスタであり、プロセッサ1により書き込まれたサイズを保持する。このサイズは、スタートアドレスからのアドレス範囲を示す。図4(b)に、サイズレジスタ403にサイズを書き込む命令の一例を示す。この命令も、図4(c)と同様に通常の転送命令(mov命令)である。なお、サイズの単位は、バイト数であっても、ライン数(キャッシュエントリー数)であってもよく、予め定められた単位であればよい。

【0047】

加算器404は、スタートアドレスレジスタ402に保持されたスタートアドレスとサイズレジスタ403に保持されたサイズとを加算する。加算結果は、アドレス範囲の終了位置を指すエンドアドレスである。加算器404は、サイズがバイト数指定の場合はバイトアドレスとして加算し、サイズがライン数指定の場合はラインアドレスとして加算すれ

ばよい。

【0048】

スタートアライナ405 a、405 bは、スタートアドレスをライン境界の位置に調整する。スタートアライナ405 aはエンドアドレスの方向に、405 bはエンドアドレスとは反対の方向に調整する。この調整によりプロセッサ1はラインサイズ及びライン境界とは無関係に任意のアドレスをスタートアドレスとして指定することができる。

【0049】

エンドアライナ406 a、406 bは、エンドアドレスをライン境界の位置に調整する。エンドアライナ406 aはスタートアドレスの方向に、406 bはスタートアドレスとは反対の方向に調整する。この調整によりプロセッサ1はラインサイズ及びライン境界とは無関係に任意の大きさを上記サイズとして指定することができる。

【0050】

図5に、スタートアライナ405 a、405 b及びエンドアライナ406 a、406 bの説明図を示す。同図において、プロセッサ1から指定されたスタートアドレスはラインNの途中の任意の位置を指す。スタートアライナ405 aは、次のライン(N+1)の先頭を指すよう調整し、調整後のアドレスをアラインスタートアドレスaとして出力する。スタートアライナ405 bは、スタートアドレスのデータを含むラインNの先頭を指すよう調整し、調整後のアドレスをアラインスタートアドレスbとして出力する。アラインスタートアドレスが指すラインをスタートラインと呼ぶ。

【0051】

また、エンドアドレスはラインMの途中の任意の位置を指す。エンドアライナ406 aは、直前のライン(M-1)の先頭を指すよう調整し、調整後のアドレスをアラインエンドアドレスaとして出力する。エンドアライナ406 bは、エンドアドレスのデータを含むラインMの先頭を指すよう調整し、調整後のアドレスをアラインエンドアドレスbとして出力する。アラインエンドアドレスが指すラインをエンドラインと呼ぶ。

【0052】

同図のように、スタートアライナ405 a及びエンドアライナ406 aはライン単位で内側アラインを行う。スタートアライナ405 b及びエンドアライナ406 bはライン単位で外側アラインを行う。さらに、ライン単位の外側アラインの後、さらに、サブライン単位の外側アラインと内側アラインが可能である。

【0053】

アクセスアドレスレジスタ407は、プロセッサ1からのメモリアクセスアドレスを保持する。

予測値レジスタ408は、初期値として、アクセスアドレスレジスタ407のメモリアクセスアドレスと定数レジスタ409の定数とを加算した値を予測値として保持し、以降、プロセッサ1がメモリアクセスを実行したときに、アクセスアドレスレジスタ407のメモリアクセスアドレスと予測値とが一致していれば、予測値と定数レジスタ409の定数とを加算した値を新たな予測値として更新し、一致していなければ、新たな初期値に更新する。

【0054】

定数レジスタ409は、プロセッサ1によるメモリアクセスアドレスのインクリメント値またはデクリメント値を保持する。このインクリメント値(またはデクリメント値)は、プロセッサ1がメモリをシーケンシャルにアクセスする場合のポストインクリメント付きロード／ストア命令のインクリメント値(またはデクリメント値)であり、例えば、図4(d)に示したコマンド中の定数として指定される。

【0055】

セクタ410は、アクセスアドレスレジスタ407のメモリアクセスアドレスと予測値レジスタ408の予測値とが一致していれば、予測値レジスタ408を選択し、一致していなければ、アクセスアドレスレジスタ407を選択する。

【0056】

加算器411は、セクタ410に選択された予測値またはメモリアクセスアドレスと、定数レジスタ409の定数とを加算する。加算後の値は、新たな予測値又は新たな初期値として予測値レジスタ408に保持される。

【0057】

比較器412は、アクセスアドレスレジスタ407のメモリアクセスアドレスと予測値レジスタ408の予測値とが一致するか否かを判定する。

実行部413は、プリフェッチ部414とタッチ部415とを備える。

【0058】

プリフェッチ部414は、プロセッサのロード命令の進行状況に応じて次にロードされるライン推定し、推定したラインをプリフェッチする予測プリフェッチを行う。

タッチ部415は、プロセッサのストア命令の進行状況に応じて次にストアされるライン推定し、推定したラインを保持するためのキャッシュエントリーを、メモリからデータをロードすることなく確保する（V＝1の設定とタグの設定とを行う）。

【0059】

<予測プリフェッチ>

図6（a）は、プリフェッチ部414による予測プリフェッチの説明図である。同図（a）において、ラインN、N＋1、・・・、N＋nはアドレスが連続するラインを示す。スタートアドレスレジスタ402及びサイズレジスタ403により定まるスタートライン、エンドラインは、それぞれラインN＋1、ラインN＋nであるものとする。また、LD（1）、LD（2）、・・・は、ロード命令によるアクセス位置を示している。

【0060】

プリフェッチ部414は、最初のロード命令LD（1）の実行時に、そのメモリアクセスアドレスに定数を加算した値を初期値として予測値レジスタ408に保持させる。2回目のロード命令LD（2）の実行時に、そのメモリアクセスアドレスと予測値レジスタ408の予測値とが一致した場合、プリフェッチ部414は、ポストインクリメント付きロード命令によるシーケンシャルアクセスであるものと推定して、次のラインN＋1をプリフェッチする。もし、次のラインがキャッシュメモリに格納されている場合は、なにもしない。

【0061】

プロセッサ1によってポストインクリメント付きロード命令によりシーケンシャルアクセスがなされた場合、予測値はメモリアクセス毎に一致していくことになる。

このようにして、予測値レジスタ408の予測値がラインNからラインN＋1のアドレスにまで更新され、メモリアクセスアドレスと一致したとき、プリフェッチ部414は、ラインN＋2をプリフェッチする。プリフェッチ部414は、この予測プリフェッチをスタートラインからエンドラインまで実行し、エンドラインのプリフェッチが完了すると予測プリフェッチを終了する。

【0062】

<予測タッチ>

図6（b）は、タッチ部415による予測タッチの説明図である。図6（a）と比較して、プロセッサ1がストア命令によりシーケンシャルにアクセスする場合に、そのメモリアクセスアドレスと予測値レジスタ408の予測値とが一致した場合に、ポストインクリメント付きストア命令によるシーケンシャルアクセスと推定し、次のラインN＋1に対応するキャッシュエントリーを確保する点が異なる。つまり、プリフェッチの代わりに、キャッシュエントリーにメモリデータをロードしないでタグ及びバリッドフラグを1に設定する点が異なっている。これ以外は予測プリフェッチと同様なので説明を省略する。

【0063】

<予測プリフェッチ処理>

図7は、プリフェッチ部414における予測プリフェッチ処理の一例を示すフローチャートである。

【0064】

同図において、プリフェッチ部414は、コマンドレジスタ401に予測プリフェッチコマンドが保持されていて（S41）、プロセッサがロード命令を実行したとき（S42）、当該メモリアクセスアドレスと定数とを加算した値を初期値として予測値レジスタ408に設定する（S43）。この最初のロード命令のメモリアクセスアドレスと予測値レジスタの予測値とは当然一致しない。予測値レジスタ408はクリアされているかランダムな値を保持しているからである。

【0065】

さらに、プロセッサがロード命令を実行し（S44）、かつメモリアクセスアドレスと予測値レジスタの予測値とが一致する場合には（S45）、プリフェッチ部414は、次ラインのラインアドレスを算出し（S46）、算出したラインアドレスがスタートラインからエンドラインまでのアドレス範囲に属していて（S47）、次ラインがキャッシュメモリにエントリーされていなければ（S48）、次ラインをプリフェッチする（S49）。

【0066】

このプリフェッチにおいて、プリフェッチ部414は、LRU方式でリプレイス対象のウェイを選択し（S401）、当該ウェイのキャッシュエントリーがダーティであればライトバックし（S402、S403）、次ラインのデータを当該キャッシュエントリーにリフィル（プリフェッチ）する（S404）。

【0067】

さらに、プリフェッチ部414は、プリフェッチしたラインがエンドラインであれば（S50）予測プリフェッチ処理を終了する。

また、上記S45において、メモリアクセスアドレスと予測値レジスタの予測値とが一致しない場合には、メモリアクセスアドレスに定数を加算した値が新たな予測値として予測値レジスタ408に設定される。また、一致しない場合は、なにもしない（メモリアクセス待ちの状態）。

【0068】

このように、プリフェッチ部414は、予測プリフェッチにおいて、プロセッサのロード命令の進行状況に応じて次のラインを推定し、推定したラインをプリフェッチするので、プロセッサ1では、キャッシュメモリのラインサイズやライン境界を認識する必要がなく、つまり、プリフェッチ用のアドレスをプロセッサ1にて管理する必要がない。キャッシュメモリにおいて推定に基づいて効率よくプリフェッチを行うことができる。

【0069】

<予測タッチ処理>

図8は、タッチ部415における予測タッチ処理の一例を示すフローチャートである。同図は、図7の予測プリフェッチ処理と比較して、S41、S42、S44、S49、S404の各ステップの代わりにS41a、S42a、S44a、S49a、S404aを有する点が異なっている。これ以外は同じなので異なる点について説明する。

【0070】

タッチ部415は、S41aにおいて、予測タッチコマンドがコマンドレジスタ401に保持されているかを判定し、S42a、S44aにおいてストア命令が実行されたか否かを判定する。また、タッチ部415は、S49aにおいてプリフェッチする代わりにタッチする。すなわち、S404aにおいて、リプレイス対象のキャッシュエントリーメモリデータをロードしないでタグ及びバリッドフラグを1に設定する。

【0071】

このように、タッチ部415は、プロセッサのストア命令の進行状況に応じて次のラインを推定し、推定したラインをタッチする（確保する）ので、プロセッサ1では、キャッシュメモリのラインサイズやライン境界を認識する必要がなく、つまりプリフェッチ用のアドレスをプロセッサ1にて管理する必要がない。キャッシュメモリにおいて推定に基づいて効率よくキャッシュエントリーをタッチすることができる。

【0072】

<変形例>

なお、本発明のキャッシュメモリは、上記の実施の形態の構成に限るものではなく、種々の変形が可能である。以下、変形例のいくつかについて説明する。

(1) 定数レジスタ409、コマンドにより設定される構成を示したが、(a) デフォルトとしてインクリメント値またはデクリメント値を保持する、(b) 複数回のロード／ストア命令における2つのメモリアクセスアドレスの差分を算出し、インクリメント値またはデクリメント値として保持する、(c) プロセッサにより指定されたアドレス方向(アドレス昇順かアドレス降順か)に応じて上記(b)を算出し保持する構成としてもよい。

(2) プリフェッチ部414において、次のラインのさらに次のライン等複数ラインをプリフェッチするようにしてもよい。同様にタッチ部415も、複数ラインをタッチするようにしてもよい。

(3) プリフェッチ部414は、比較器412で複数回一致した場合にプリフェッチを開始するようにしてもよい。タッチ部415も同様である。

(4) 上記実施の形態では、4ウェイ・セット・アソシエイティブのキャッシュメモリを例に説明したが、ウェイ数は、いくつでもよい。また、上記実施の形態では、セット数が16である例を説明したが、セット数はいくつでもよい。

(5) 上記実施の形態では、セット・アソシエイティブのキャッシュメモリを例に説明したが、フル・アソシエイティブ方式やダイレクトマップ方式のキャッシュメモリであってもよい。

(6) 上記実施の形態では、サブラインのサイズをラインのサイズの1/4としているが、1/2、1/8、1/16等他のサイズでもよい。その場合、各キャッシュエントリーは、サブラインと同数のバリッドフラグおよびダーティフラグをそれぞれ保持すればよい。

(7) 上記実施の形態におけるサイズレジスタ403は、サイズを保持する代わりにプリフェッチすべき回数を保持する構成としてもよい。この場合には、予測処理部39は実際にプリフェッチした回数をカウントし、カウント値がプリフェッチすべき回数に達したとき、プリフェッチを停止する構成とすればよい。このようにプリフェッチ回数をカウントすることにより、プロセッサにおけるプログラム進行状態を監視することができる。

【0073】

(実施の形態2)

第1の実施の形態では、説明した予測プリフェッチ及び予測タッチでは今後アクセスするであろうラインを推定する場合の構成を説明した。本実施の形態では、既にアクセスしたであろうラインを推定して予測クリーニング(ライトバック)や予測ウィーク化(アクセス順位の最弱化)や予測無効化も行いう構成について説明する。

【0074】

<キャッシュメモリの構成>

図9は、本発明の実施の形態2におけるキャッシュメモリの構成を示すブロック図である。同図のキャッシュメモリは、図2に示した構成と比較して、ウェイ31a~31dの代わりにウェイ131a~131dを備える点と、制御部38の代わりに制御部138を備える点とが異なっている。以下、同じ点は説明を省略して、異なる点を中心に説明する。

【0075】

ウェイ131aは、ウェイ31aと比べて、各キャッシュエントリー中に、Cフラグ、Wフラグ及びUフラグが追加されている点と、ライン単位のバリッドフラグVの代わりにサブライン毎のバリッドフラグV0~V3を有する点と、ライン単位のダーティフラグDの代わりにサブライン毎のダーティフラグD0~D3を有する点とが異なる。ウェイ131b~131dも同様である。

【0076】

図10に、キャッシュエントリーのビット構成を示す。1つのキャッシュエントリーは、バリッドフラグV0~V3、21ビットのタグ、128バイトのラインデータ、ウィー

クフラグW、使用フラグU及びダーティフラグD 0～D 3を保持する。

【0077】

タグは21ビットのタグアドレスのコピーである。

ラインデータは、タグアドレスおよびセットインデックスにより特定されるブロック中の128バイトデータのコピーであり、32バイトの4つのサブラインからなる。

【0078】

バリッドフラグV 0～V 3は、4つのサブラインに対応し、サブラインが有効か否かを示す。

Cフラグ（クリーニングフラグ）は、キャッシングを終了してよいかどうかを示すキャッシング終了属性を示す。C=0は、以降に書き込みがなされる可能性があることを意味する。C=1は、以降に書き込みがなされないことを意味し、ダーティであればクリーニング（ライトバック）によりキャッシングを終了すべきであることを意味する。

【0079】

ウィークフラグWは、プロセッサからのアクセスに関しては、これ以上使用するか否かを示し、キャッシュメモリにおけるリプレース制御に関しては、他のキャッシュエントリーよりも先に追い出してもよい最弱のリプレース対象であることを示す。

【0080】

使用フラグUは、そのキャッシュエントリーにアクセスがあったか否かを示し、LRU方式におけるキャッシュエントリー間のアクセス順序データの代わりに用いられる。より正確には、使用フラグUの1は、アクセスがあったことを、0はないことを意味する。ただし、1つのセット内の4つウェイの使用フラグが全て1になれば、0にリセットされる。別言すれば、使用フラグUは、アクセスされた時期が古いか新しいか2つの相対的な状態を示す。つまり、使用フラグUが1のキャッシュエントリーは、使用フラグが0のキャッシュエントリーよりも新しくアクセスされたことを意味する。

【0081】

ダーティフラグD 0～D 3は、4つのサブラインに対応し、そのサブラインにプロセッサから書き込みがあったか否か、つまりサブライン中にキャッシュされたデータが存在するが書き込みによりメモリ中のデータと異なるためメモリに書き戻すことが必要か否かを示す。

【0082】

制御部138は、制御部38と比べて、予測処理部39の代わりに予測処理部139を有する点が異なり、予測によりCフラグを設定する点と、予測によりWフラグを設定する点と、LRU方式におけるアクセス順序情報の代わりに使用フラグUを用いる点とが異なる。

【0083】

<使用フラグUの説明>

図11は、制御部138による使用フラグの更新例を示す。同図の上段、中段、下段は、ウェイ0～3に跨るセットNを構成する4つのキャッシュエントリーを示している。4つのキャッシュエントリー右端の1又は0は、それぞれ使用フラグの値である。この4つの使用フラグUをU 0～U 3と記す。

【0084】

同図上段では（U 0～U 3）＝（1、0、1、0）であるので、ウェイ0、2のキャッシュエントリーはアクセスがあったことを、ウェイ1、3のキャッシュエントリーはアクセスがないことを意味する。

【0085】

この状態で、メモリアクセスがセットN内のウェイ1のキャッシュエントリーにヒットした場合、同図中段に示すように、（U 0～U 3）＝（1、1、1、0）に更新される。つまり、実線に示すようにウェイ1の使用フラグU1が0から1に更新される。

【0086】

さらに、同図中段の状態で、メモリアクセスがセットN内のウェイ3のキャッシュエ

トリーにヒットした場合、同図下断に示すように、 $(U0 \sim U3) = (0, 0, 0, 1)$ に更新される。つまり、実線に示すようにウェイ3の使用フラグU1が0から1に更新される。加えて、破線に示すようにウェイ3以外の使用フラグU0～U2が1から0に更新される。これにより、ウェイ3のキャッシュエントリーが、ウェイ0～2の各キャッシュエントリーよりも新しくアクセスされたことを意味することになる。

【0087】

制御部138は、キャッシュミス時に $W=1$ のキャッシュエントリーが存在しなければ、使用フラグに基づいてリプレース対象のキャッシュエントリーを決定してリプレースを行う。例えば、制御部138は、図5上段では、ウェイ1とウェイ3の何れかをリプレース対象と決定し、図5中段ではウェイ3をリプレース対象と決定し、図5下段ではウェイ0～2の何れかをリプレース対象と決定する。

【0088】

＜ウィークフラグWの説明＞

図12(a)ウィークフラグが存在しないと仮定した場合の比較例であり、キャッシュエントリーがリプレースされる様子を示す図である。同図においても、図11と同様にウェイ0～3に跨るセットNを構成する4つのキャッシュエントリーを示している。、4つのキャッシュエントリー右端の1又は0は、それぞれ使用フラグの値である。また、データEのみアクセス頻度の低いデータを、データA、B、C、Dはアクセス頻度の高いデータとする。

【0089】

同図(a)の第1段目の状態で、プロセッサ1がデータEにアクセスすると、キャッシュミスが発生する。このキャッシュミスにより、例えば、 $U=0$ のキャッシュエントリーの中からアクセス頻度の高いデータCのキャッシュエントリーがアクセス頻度の低いデータEにリプレースされ、第2段目の状態となる。

【0090】

第2段目の状態で、プロセッサ1がデータCにアクセスすると、キャッシュミスが発生する。このキャッシュミスにより、 $U=0$ のキャッシュエントリーであるアクセス頻度の高いデータDのキャッシュエントリーがアクセス頻度の高いデータCにリプレースされ、第3段目の状態となる。

【0091】

第3段目の状態で、プロセッサ1がデータDにアクセスすると、キャッシュミスが発生する。このキャッシュミスにより、例えば、アクセス頻度の高いデータCのキャッシュエントリーがアクセス頻度の高いデータDにリプレースされ、第3段目の状態となる。

【0092】

同様に、第4段目でも、使用頻度の低いデータEはリプレース対象として選択されないで、キャッシュメモリーに残っている。

第5段目の状態で、使用頻度の低いデータEは最も古い($U=0$)であることから、リプレース対象として選択されて、追い出される。

【0093】

このように、擬似LRU方式において(通常のLRU方式においても)、アクセス頻度の低いデータEによって、4ウェイの場合は最悪4回のキャッシュミスを誘発する場合がある。

【0094】

図12(b)は、リプレース処理におけるウィークフラグWの役割を示す説明図である。

同図(b)の第1段目の状態(同図(a)の第1段目と同じ)で、プロセッサ1がデータEにアクセスすると、キャッシュミスが発生する。このキャッシュミスにより、例えば、 $U=0$ のキャッシュエントリーの中からアクセス頻度の高いデータCのキャッシュエントリーがアクセス頻度の低いデータEにリプレースされる。このとき、プロセッサ1は、データEのキャッシュエントリーにウィークフラグWを1に設定するものとする。これに

より、次のキャッシュミス時にデータEのキャッシュエントリーが真っ先に追い出され、第2段目の状態となる。

【0095】

第2段目の状態で、プロセッサ1がデータCにアクセスすると、キャッシュミスが発生する。このキャッシュミスにより、 $W=1$ のキャッシュエントリーであるアクセス頻度の低いデータEのキャッシュエントリーがリプレース対象として選択され、アクセス頻度の高いデータCにリプレースされ、第3段目の状態となる。

【0096】

このように、ウィークフラグWを設けることにより、アクセス頻度の低いデータによるキャッシュミスの誘発を低減することができる。

【0097】

<予測処理部の構成>

図13は、予測処理部139の構成を示すブロック図である。同図の予測処理部139は、図3に示した予測処理部39と比較して、実行部413の代わりに実行部413aを備える点が異なっている。実行部413aは、実行部413と比べて、C設定部416、W設定部417が追加されている。

【0098】

C設定部416は、プロセッサのストア命令の進行状況に応じてストアが完了した直前のラインを推定し、推定したラインのCフラグを1に設定する。

W設定部417は、プロセッサのストア命令の進行状況に応じてストアが完了した直前のラインを推定し、推定したラインのWフラグを1に設定する。

【0099】

<Cフラグ設定処理の説明図>

図14(a)は、C設定部416によるCフラグ設定処理の説明図である。同図(a)において、ラインN、N+1、・・・、N+n、スタートライン、エンドラインは、図6(a)と同様である。

【0100】

C設定部416は、プロセッサ1がシーケンシャルにラインN、N+1、・・・に対してデータをストアしていく場合に、例えば、ラインN+1に対してストア命令ST(1)、ST(2)が実行され、ST(2)のメモリアクセスアドレスと予測値とが一致した場合に、ラインNに対するシーケンシャルアクセスによるストアが完了したものと推定し、ラインNのCフラグを1に設定する。Cフラグが1に設定されたラインNは、キャッシュミスの発生を待たずにクリーニング処理によりライトバックされる。

【0101】

同様、ラインN+2へのストア命令実行中に予測値を一致した場合、ラインN+1に対するシーケンシャルアクセスによるストアが完了したものと推定し、ラインN+1のCフラグを1に設定する。

【0102】

このように、C設定部416は、Cフラグの設定をスタートラインからエンドラインまでの範囲内で実行する。

【0103】

<Wフラグ設定処理の説明図>

図14(b)は、W設定部417によるWフラグ設定処理の説明図である。同図(a)と比べて、Cフラグの代わりにWフラグを設定する点のみが異なり、これ以外は同様であるので説明を省略する。

【0104】

Wフラグが1に設定されたラインは、キャッシュミスの発生時に真っ先にリプレース対象として選択され、キャッシュメモリから追い出される。

【0105】

<Cフラグ設定処理フロー>

図 1 5 は、C 設定部 4 1 6 における C フラグ設定処理の一例を示すフローチャートである。

【0 1 0 6】

同図において、C 設定部 4 1 6 は、コマンドレジスタ 4 0 1 に C フラグ設定コマンドが保持されていて (S 4 1 b)、プロセッサがストア命令を実行したとき (S 4 2 b)、当該メモリアクセスアドレスと定数とを加算した値を初期値として予測値レジスタ 4 0 8 に設定する (S 4 3)。この最初のストア命令のメモリアクセスアドレスと予測値レジスタの予測値とは当然一致しない。予測値レジスタ 4 0 8 はクリアされているかランダムな値を保持しているからである。

【0 1 0 7】

さらに、プロセッサがストア命令を実行し (S 4 4 b)、かつメモリアクセスアドレスと予測値レジスタの予測値とが一致する場合には (S 4 5)、C 設定部 4 1 6 は、直前のラインのラインアドレスを算出し (S 4 6 b)、算出したラインアドレスがスタートラインからエンドラインまでのアドレス範囲に属していて (S 4 7)、直前のラインがキャッシュメモリにエントリされているか (S 4 8 b)、直前のラインの C フラグを 1 に設定する (S 4 9 b)。

【0 1 0 8】

さらに、C 設定部 4 1 6 は、直前のラインがエンドラインであれば (S 5 0) C フラグ設定処理を終了する。

このように、C 設定部 4 1 6 は、プロセッサのストア命令の進行状況に応じて、ストアが完了した直前のラインを推定し、推定したラインの C フラグを 1 に設定するので、プロセッサ 1 では、キャッシュメモリのラインサイズやライン境界を認識する必要がなく、つまり、クリーニングしてもよいラインをラインアドレスとしてプロセッサ 1 にて管理する必要がない。キャッシュメモリにおいて推定に基づいて効率よくクリーニングを行うことができる。

【0 1 0 9】

< W フラグ設定処理フロー >

図 1 6 は、W 設定部 4 1 7 における W フラグ設定処理の一例を示すフローチャートである。同図のフローは、図 1 5 の C フラグ設定処理と比較して、C フラグの代わりに W フラグを設定する点のみが異なり、これ以外は同様であるので説明を省略する。

【0 1 1 0】

このように、W 設定部 4 1 7 は、プロセッサのストア命令の進行状況に応じて、ストアが完了した直前のラインを推定し、推定したラインの W フラグを 1 に設定するので、キャッシュメモリにおいて推定に基づいてストアが完了した直前のラインを効率よくリプレイス対象とすることができる。その際、プロセッサ 1 では、キャッシュメモリのラインサイズやライン境界を認識する必要がなく、つまり、リプレイスしてもよいラインをラインアドレスとしてプロセッサ 1 にて管理する必要がない。

【0 1 1 1】

< クリーニング処理 >

図 1 7 は、制御部 1 3 8 におけるクリーニング処理の一例を示すフローチャートである。

【0 1 1 2】

同図のように、制御部 1 3 8 は、ループ 1 の処理 (S 9 0 0 ~ S 9 1 3) において、セットインデックス (S I) 0 ~ 1 5 を順に指定する (S 9 0 1) ことにより、1 6 個のすべてのセットに対してループ 2 の処理を行う。ループ 2 の処理 (S 9 0 0 ~ S 9 1 3) において、制御部 1 3 8 は、セット内のウェイ 0 ~ 3 の C フラグを読み出す (S 9 0 3) ことにより、C = 1 のキャッシュエントリを探索する (S 9 0 4)。ループ 3 の処理 (S 9 0 5 ~ 9 1 0) において、制御部 1 3 8 は、C = 1 のキャッシュエントリに対して、サブライン単位のダーティフラグを読み出し (S 9 0 6)、ダーティであれば (S 9 0 7)、そのサブラインのデータをメモリ 2 に書き戻し (S 9 0 8)、当該ダーティフラグを 0

にリセットする（S 9 0 9）。このサブラインデータの書き戻しにおいて、制御部 1 3 8 は、ループ 4 の処理（S 9 2 0 ～ S 9 2 3）のように、空きサイクルにおいて（S 9 2 0）、1ワードずつ書き戻す（S 9 2 2）。

このように、制御部 1 3 8 は、全てのキャッシュエントリーの C フラグを順にチェックして、C = 1 のキャッシュエントリーを探索し、ダーティであればキャッシュメモリ 3 からメモリ 2 に書き戻す。

【0 1 1 3】

このように、クリーニング処理では、これ以上書き込みされないことを示す C フラグを有するキャッシュエントリーを、キャッシュミスが発生する前にライトバックするので、キャッシュミス時にはロードペナルティが発生するだけでライトバックペナルティの発生を低減することができる。これによりキャッシュメモリの効率を向上させ、アクセス速度を向上させることができる。

【0 1 1 4】

< U フラグ更新処理 >

図 1 8 は、制御部 1 3 8 における U フラグ更新処理を示すフローチャートである。同図では、バリッドフラグが 0（無効）であるキャッシュエントリーの使用フラグ U は 0 に初期化されているものとする。

【0 1 1 5】

同図において、制御部 1 3 8 は、キャッシュヒットしたとき（ステップ S 6 1）、セットインデックスにより選択されたセットにおけるヒットしたウェイの使用フラグ U を 1 にセットし（ステップ S 6 2）、そのセット内の他のウェイの使用フラグ U を読み出し（ステップ S 6 3）、読み出した使用フラグ U が全て 1 であるか否かを判定し（ステップ S 6 4）、全て 1 でなければ終了し、全て 1 であれば他のウェイの全ての使用フラグ U を 0 にリセットする（ステップ S 6 5）。

【0 1 1 6】

このようにして制御部 1 3 8 は、図 1 1、図 1 2（a）（b）に示した更新例のように、使用フラグ U を更新する。

【0 1 1 7】

< リプレース処理 >

図 1 9 は、制御部 1 3 8 におけるリプレース処理を示すフローチャートである。同図において制御部 1 3 8 は、メモリアクセスがミスしたとき（ステップ S 9 1）、セットインデックスにより選択されたセットにおける、4 つウェイの使用フラグ U 及びウィークフラグ W を読み出し（ステップ S 9 2）、W = 1 のウェイが存在するか否かを判定する（ステップ S 9 3）。W = 1 のウェイが存在しないと判定された場合、U = 0 のウェイを 1 つ選択する（ステップ S 9 4）。このとき、使用フラグ U が 0 になっているウェイが複数存在する場合は、制御部 1 3 8 はランダムに 1 つを選択する。また、W = 1 のウェイが存在すると判定された場合、U フラグの値に関わらず W = 1 のウェイを 1 つ選択する（ステップ S 9 5）。このとき、ウィークフラグ W が 1 になっているウェイが複数存在する場合は、制御部 1 3 8 はランダムに 1 つを選択する。

【0 1 1 8】

さらに、制御部 1 3 8 は、当該セットにおける選択されたウェイのキャッシュエントリーを対象にリプレースし（ステップ S 9 6）、リプレース後に当該キャッシュエントリーの使用フラグ U を 1 に、ウィークフラグ W を 0 初期化する（ステップ S 9 7）。なお、このときバリッドフラグ V、ダーティフラグ D は、それぞれ 1、0 に初期化される。

【0 1 1 9】

このように、W = 1 のウェイが存在しない場合、リプレース対象は、使用フラグ U が 0 のキャッシュエントリーの中から 1 つ選択される。

また、W = 1 のウェイが存在する場合、リプレース対象は、使用フラグ U が 0 であると 1 であるとを問わず、W = 1 のウェイのキャッシュエントリーから 1 つ選択される。これにより図 1 4（a）（b）に示したように、アクセス頻度の低いデータがキャッシュメモ

リに残ることによるキャッシュミスの誘発を低減することができる。

【0120】

以上説明してきたように、本実施の形態におけるキャッシュメモリによれば、プロセッサ1によるストア命令の進行状態から、ストアが完了してラインを推定し、推定したラインに対してCフラグ又はWフラグを設定するので、ストアが完了した直前のラインを効率よくリプレース対象を指定することができる。その際、プロセッサ1によってキャッシュメモリのライン境界やラインサイズを管理する必要がなく、キャッシュ管理のための負荷を小さくすることができる。

【0121】

また、ウィークフラグ $W=1$ でかつダーティフラグ $=1$ のラインを、プロセッサからこれ以上書き込みがなされないラインとして、クリーニングすることにより、キャッシュミス時のライトバックペナルティを低減することができる。

【0122】

また、これ以上使用されないキャッシュエントリーに $W=1$ が設定され、 $W=1$ のキャッシュエントリーが真っ先にリプレース対象として選択されるので、アクセス頻度の低いデータがキャッシュメモリに残ることによるキャッシュミスの誘発を低減することができる。

【0123】

また、従来のLRU方式におけるアクセス順序を示すデータの代わりに1ビットの使用フラグを用いる擬似LRU方式を採用することにより、アクセス順序データとして1ビットのフラグでよいので、アクセス順序データのデータ量が少ないこと及び更新が簡単であることからハードウェア規模を小さくすることができる。

【0124】

＜変形例＞

(1) 図4(a)(b)(c)に示した各命令は、コンパイラによりプログラム中に挿入してもよい。その際、コンパイラは、例えば配列データの書き込みや、圧縮動画データをデコードする際のブロックデータの書き込み等、これ以上書き込みをしないプログラム位置に、上記各命令を挿入するようにすればよい。

(2) 上記クリーニング処理において無効化処理を行う構成としてもよい。すなわち、図17に示したフローチャートにおいて、S907にてダーティでないと判定された場合、さらに、当該キャッシュエントリーを無効化(Vフラグをリセット)するステップを追加してもよい。さらに、クリーニング処理におけるライトバックの後に無効化するステップを追加してもよい。

【産業上の利用可能性】

【0125】

本発明は、メモリアクセスを高速化するためのキャッシュメモリに適しており、例えば、オンチップキャッシュメモリ、オフチップキャッシュメモリ、データキャッシュメモリ、命令キャッシュメモリ等に適している。

【図面の簡単な説明】

【0126】

【図1】本発明の実施の形態1におけるプロセッサ、キャッシュメモリ、メモリを含むシステムの概略構成を示すブロック図である。

【図2】キャッシュメモリの構成例を示すブロック図である。

【図3】予測処理部の構成例を示すブロック図である。

【図4】(a) スタートアドレスレジスタにスタートアドレスを書き込む命令の一例を示す。(b) サイズレジスタにサイズを書き込む命令の一例を示す。(c) コマンドレジスタにコマンドを書き込む命令の一例を示す。(d) コマンドの一例を示す。

【図5】スタートアライナ及びエンドアライナの説明図である。

【図6】(a) プリフェッチ部による予測プリフェッチの説明図である。(b) タッチ部による予測タッチの説明図である。

【図 7】 プリフェッチ部における予測プリフェッチ処理の一例を示すフローチャートである。

【図 8】 タッチ部における予測タッチ処理の一例を示すフローチャートである。

【図 9】 本発明の実施の形態 2 におけるキャッシュメモリの構成を示すブロック図である。

【図 10】 キャッシュエントリーのビット構成を示す。

【図 11】 制御部による使用フラグの更新例を示す。

【図 12】 (a) ウィークフラグが存在しない場合にキャッシュエントリーがリプレイスされる様子を示す図である。(b) リプレイス処理におけるウィークフラグ W の役割を示す説明図である。

【図 13】 予測処理部の構成を示すブロック図である。

【図 14】 (a) C 設定部による C フラグ設定処理の説明図である。(b) W 設定部による W フラグ設定処理の説明図である。

【図 15】 C フラグ設定部における C フラグ設定処理の一例を示すフローチャートである。

【図 16】 W フラグ設定部における W フラグ設定処理の一例を示すフローチャートである。

【図 17】 制御部におけるクリーニング処理の一例を示すフローチャートである。

【図 18】 制御部における U フラグ更新処理を示すフローチャートである。

【図 19】 制御部におけるリプレイス処理を示すフローチャートである。

【符号の説明】

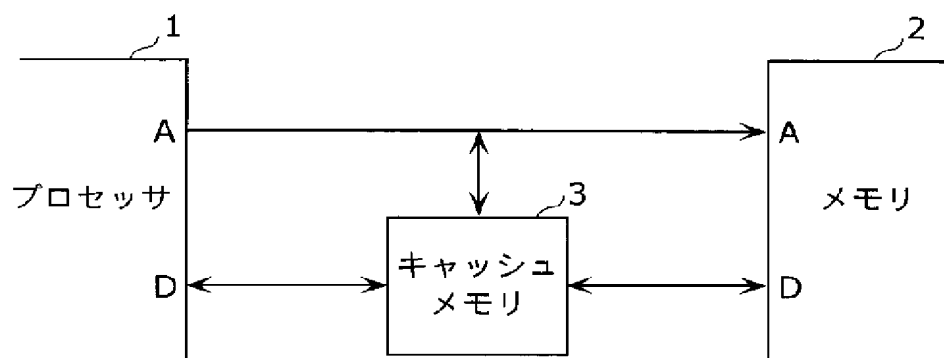
【0127】

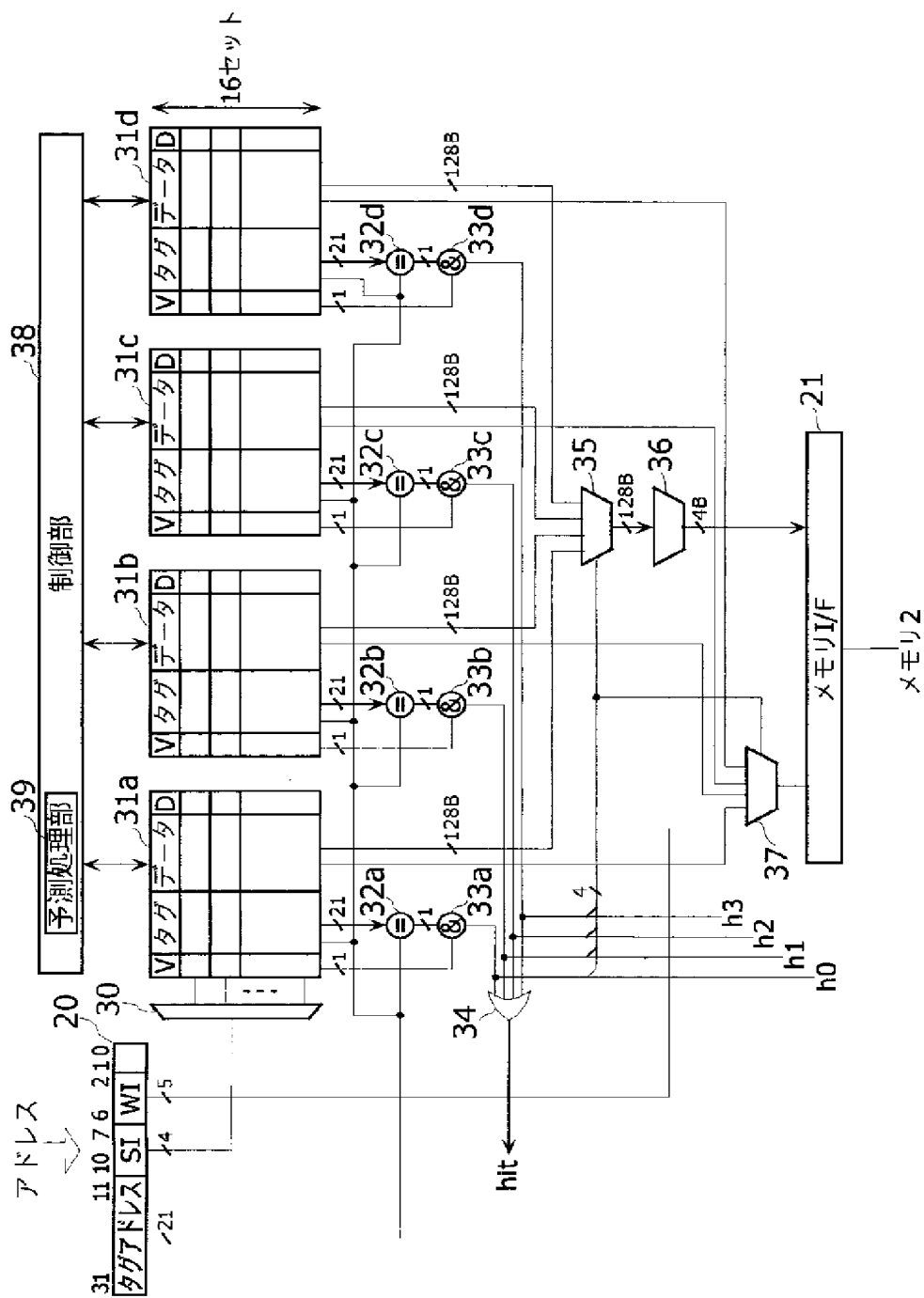
1	プロセッサ
2	メモリ
3	キャッシュメモリ
20	アドレスレジスタ
21	メモリ I/F
30	デコーダ
31 a ~ 31 d	ウェイ
32 a ~ 32 d	比較器
33 a ~ 33 d	アンド回路
34	オア回路
35	セレクタ
36	セレクタ
37	デマルチプレクサ
38	制御部
39	予測処理部
131 a	ウェイ
131 a ~ 131 d	ウェイ
131 b ~ 131 d	ウェイ
138	制御部
139	予測処理部
401	コマンドレジスタ
402	スタートアドレスレジスタ
403	サイズレジスタ
404	加算器
405 a、405 b	スタートアライナ
406 a、406 b	エンドアライナ
407	アクセスアドレスレジスタ
408	予測値レジスタ

4 0 9	定数レジスタ
4 1 0	セクタ
4 1 1	加算器
4 1 2	比較器
4 1 3	実行部
4 1 3 a	実行部
4 1 4	プリフェッチ部
4 1 5	タッチ部
4 1 6	C設定部
4 1 7	W設定部

【書類名】 図面

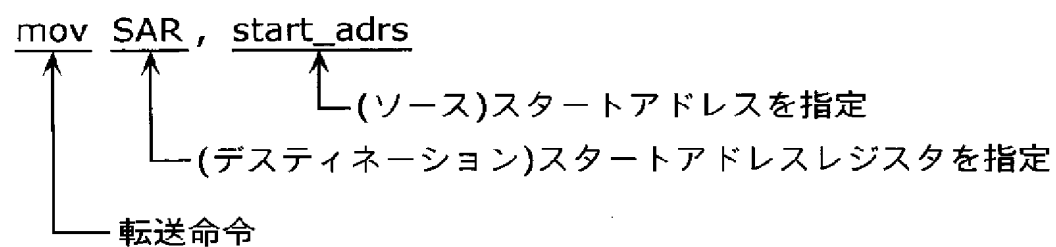
【図 1】



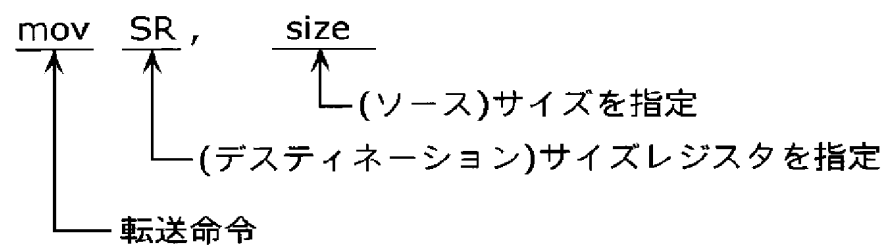


【図 4】

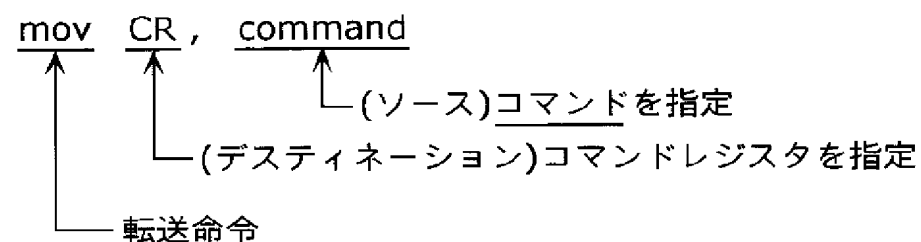
(a)



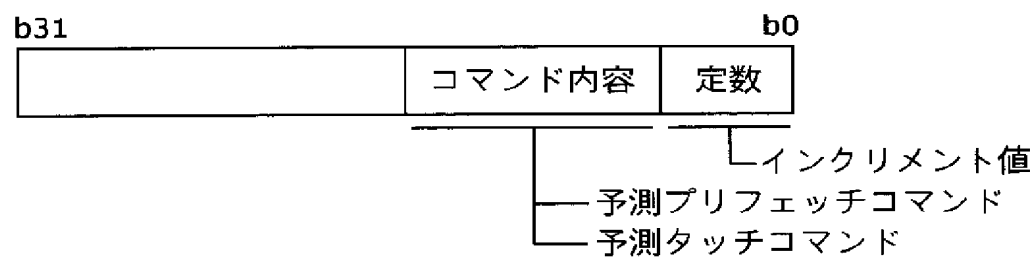
(b)

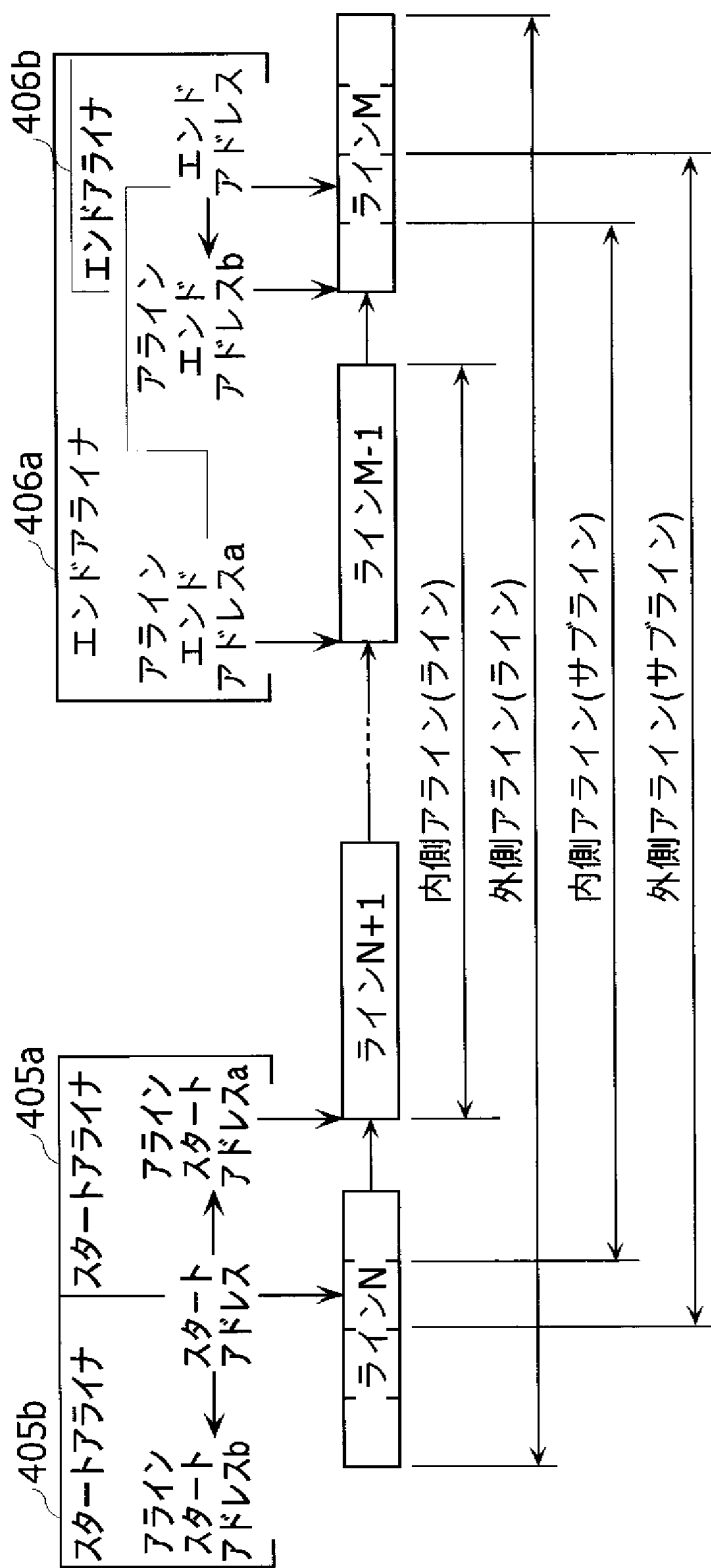


(c)

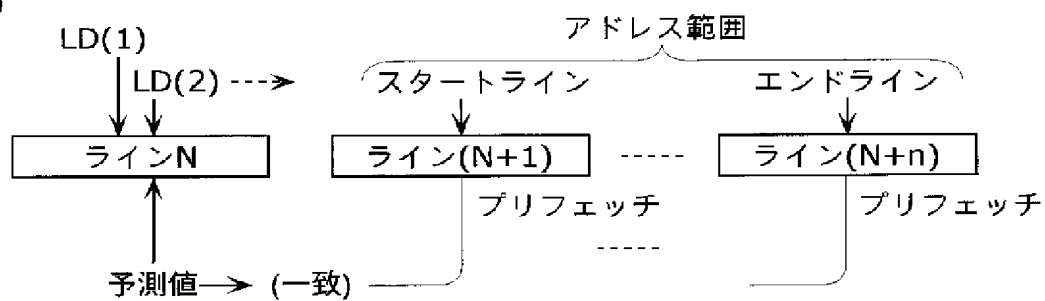


(d)

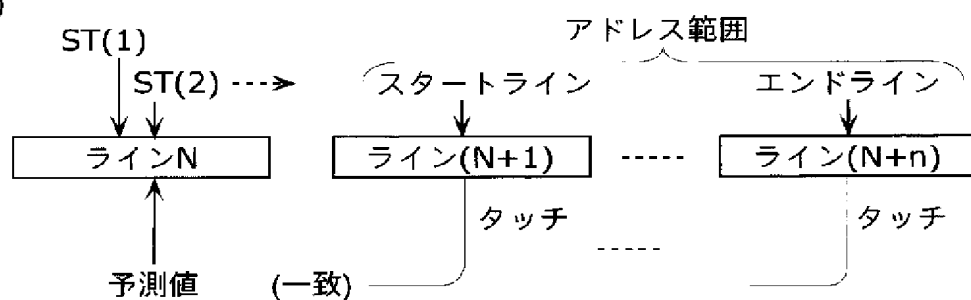




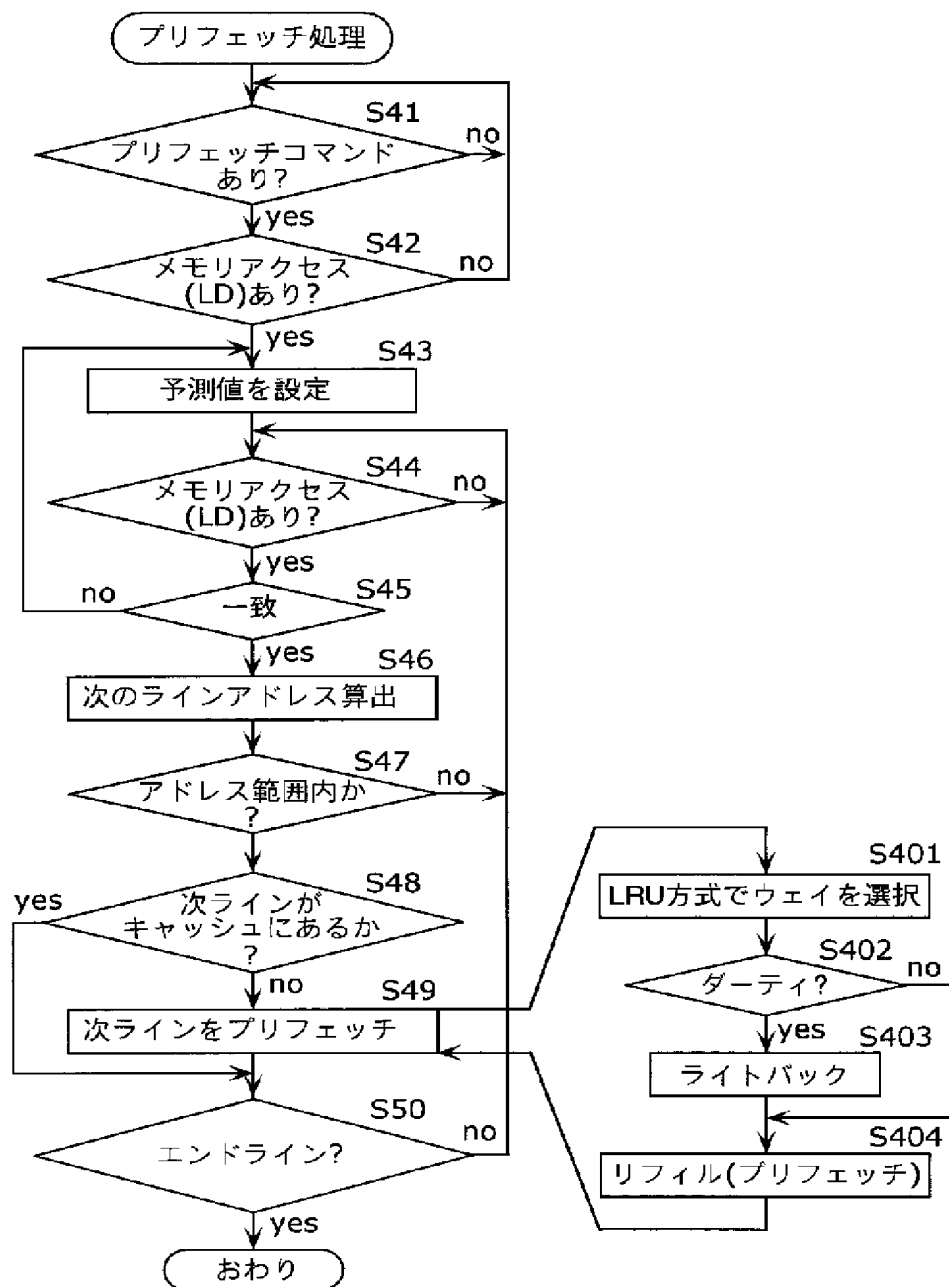
(a)



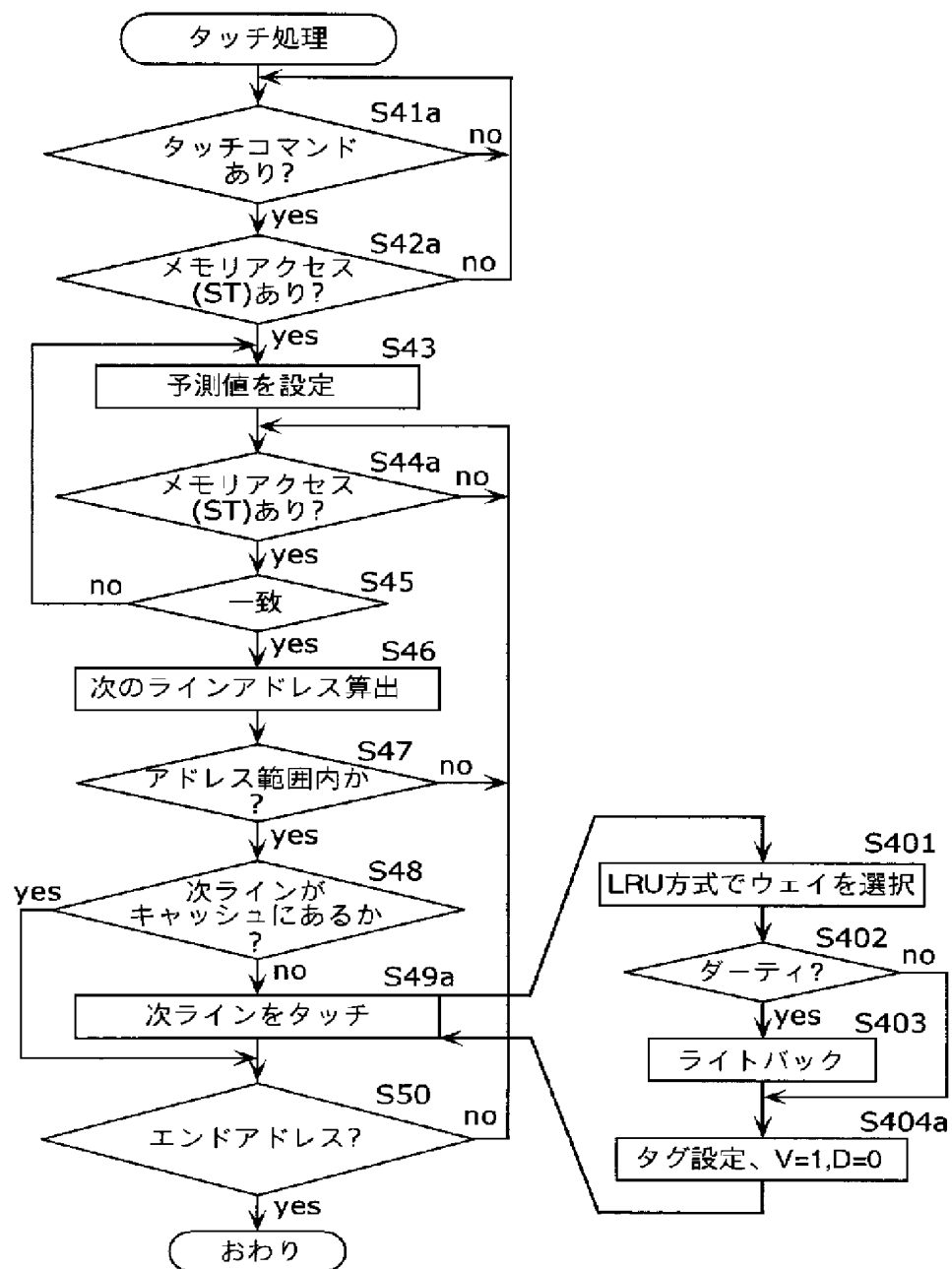
(b)

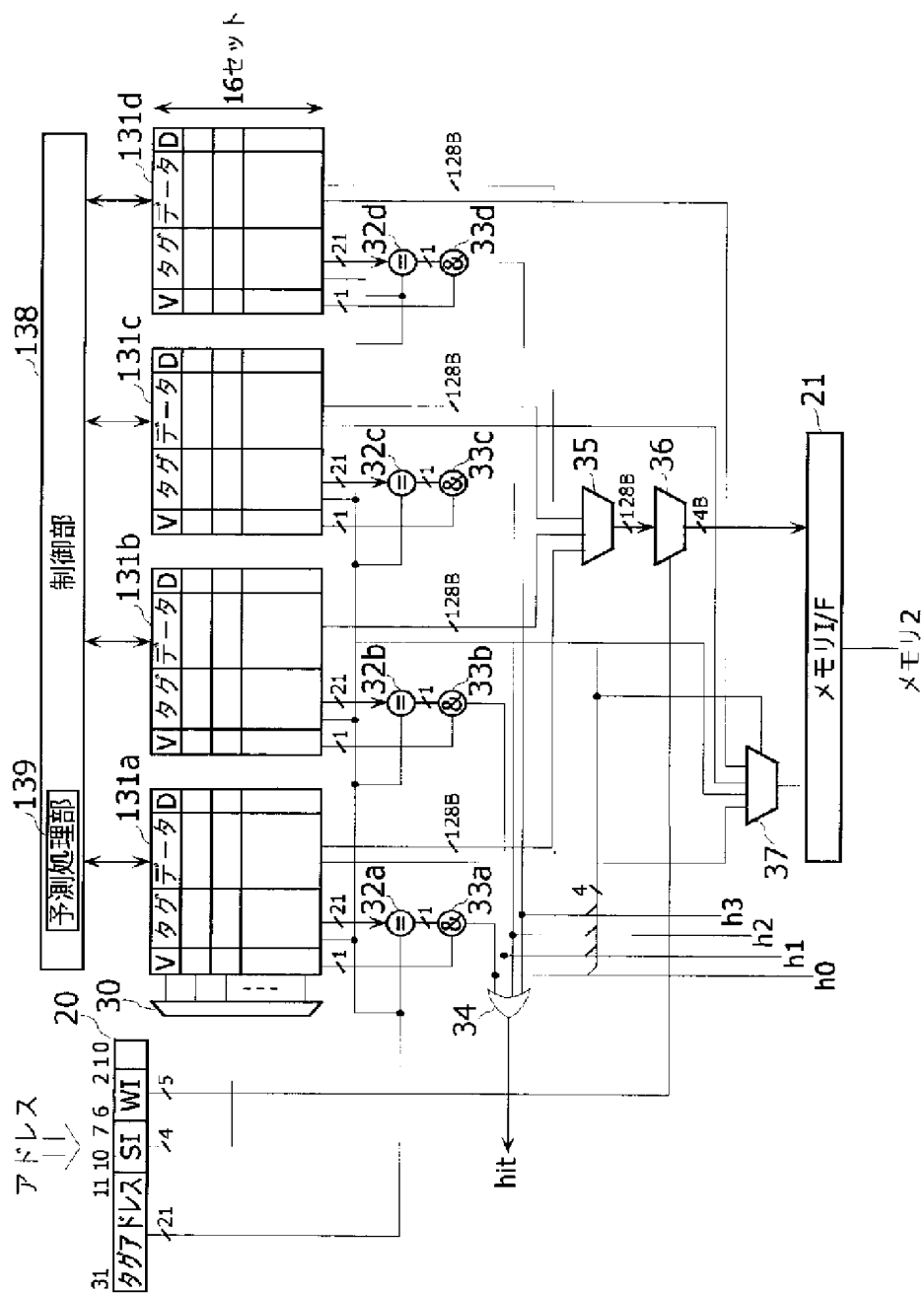


【図 7】

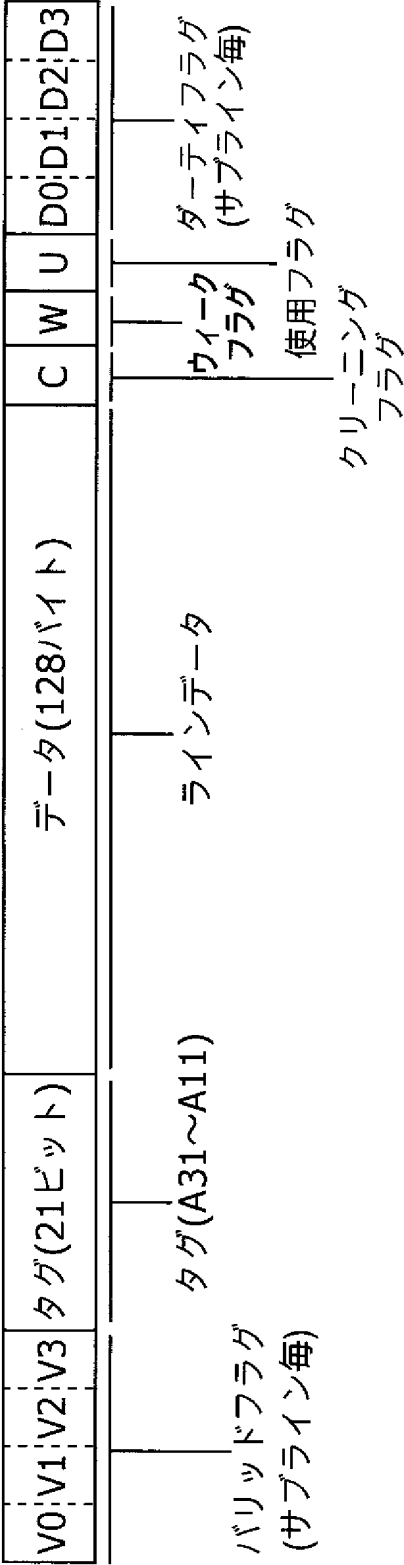


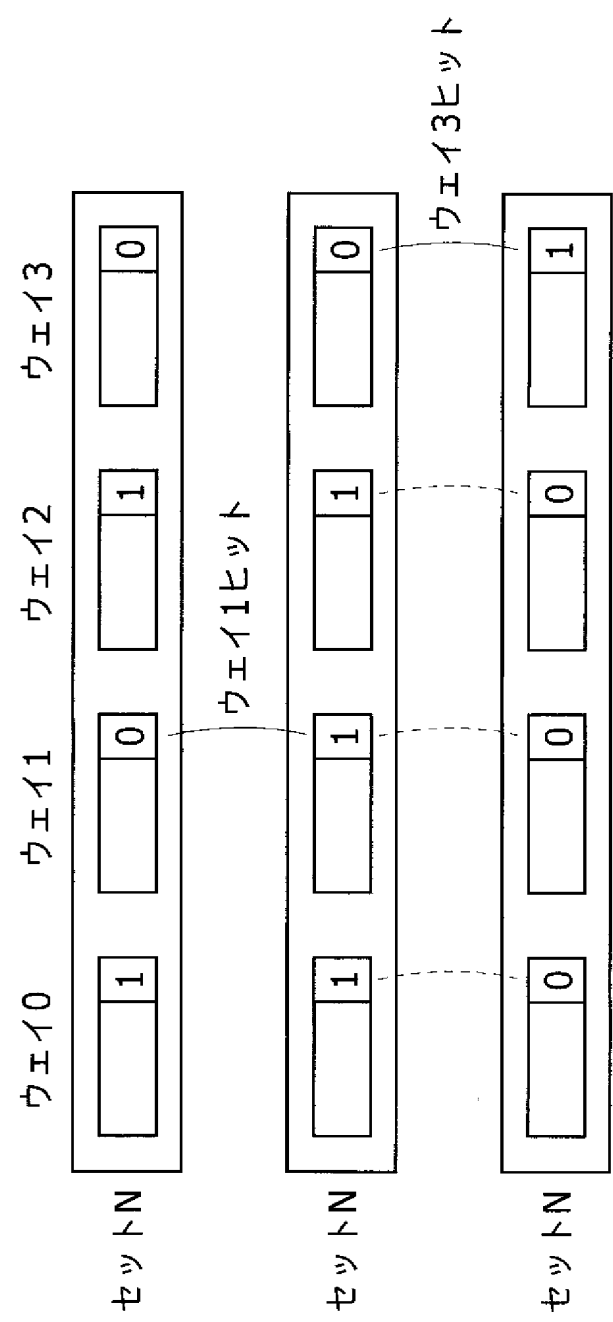
【図 8】



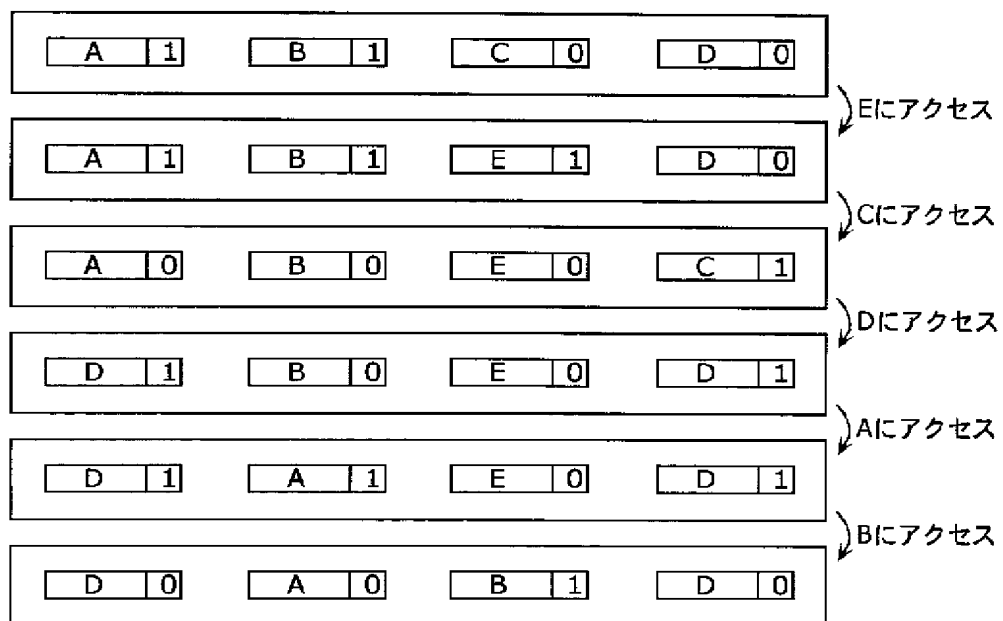


サブライン サブライン サブライン サブライン

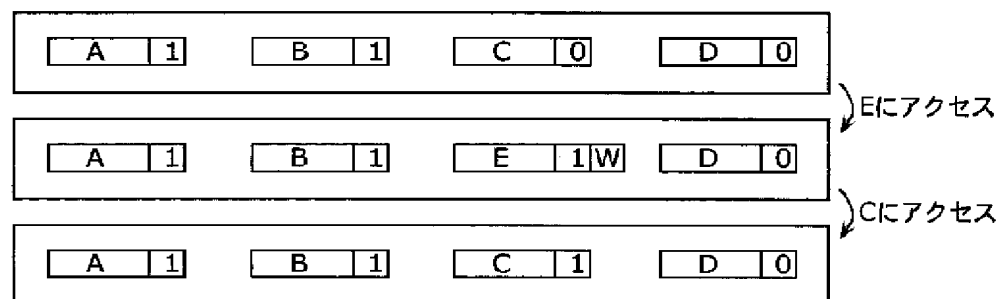


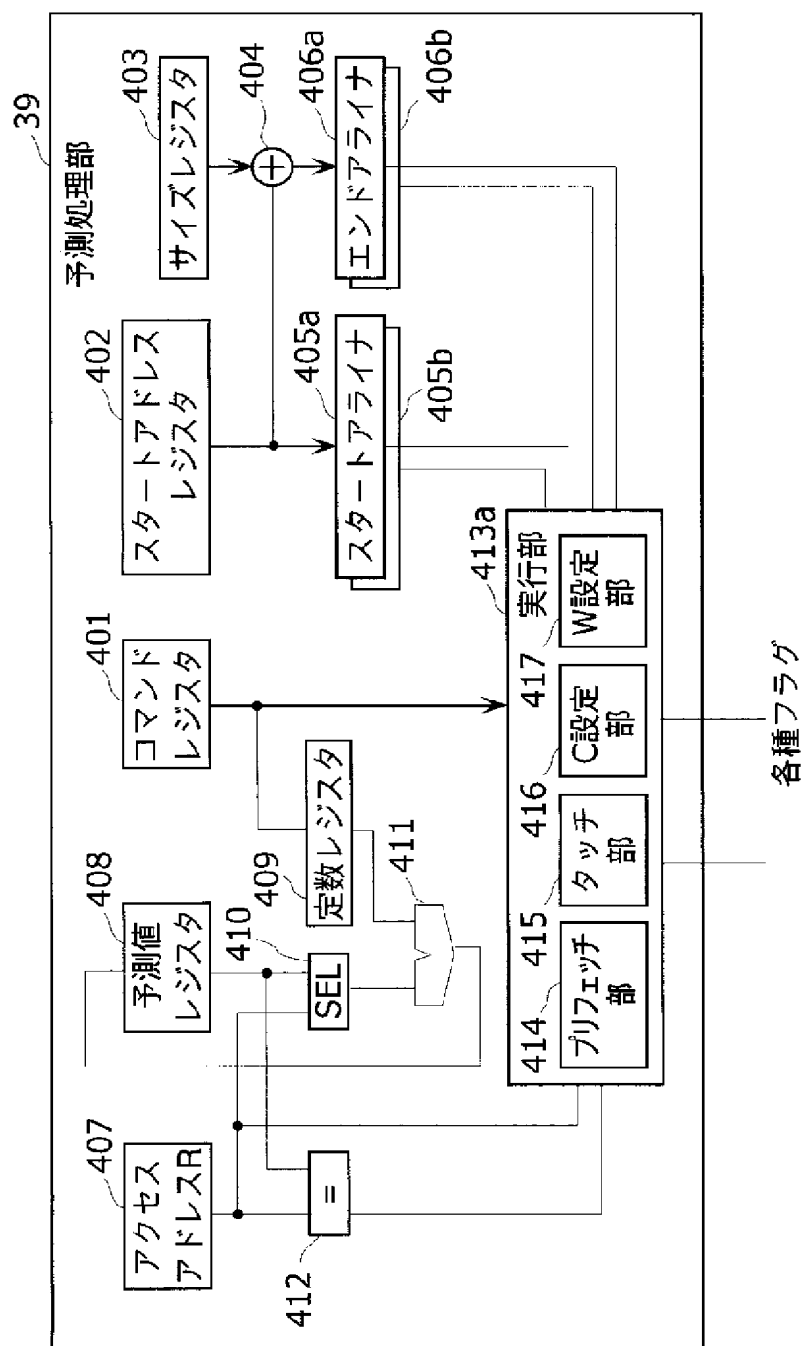


(a)

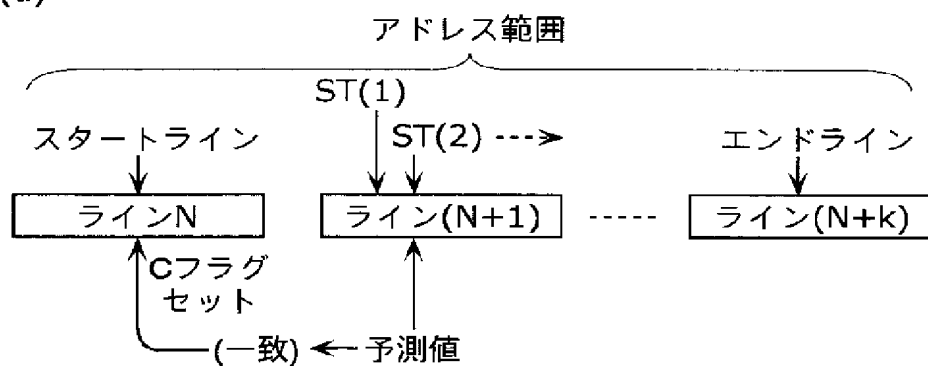


(b)

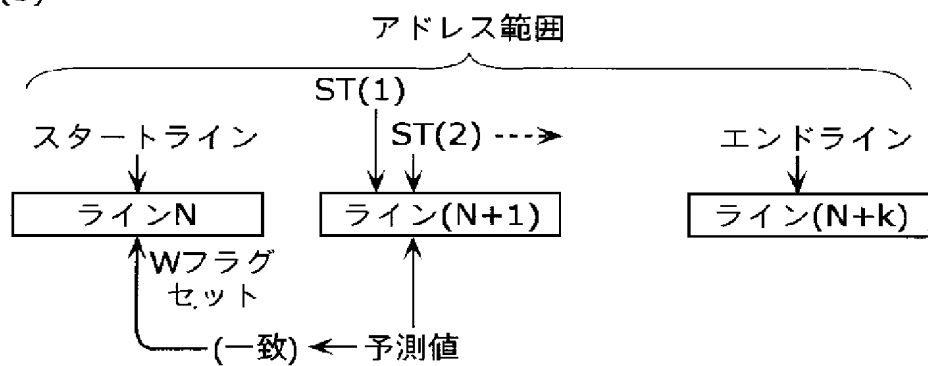


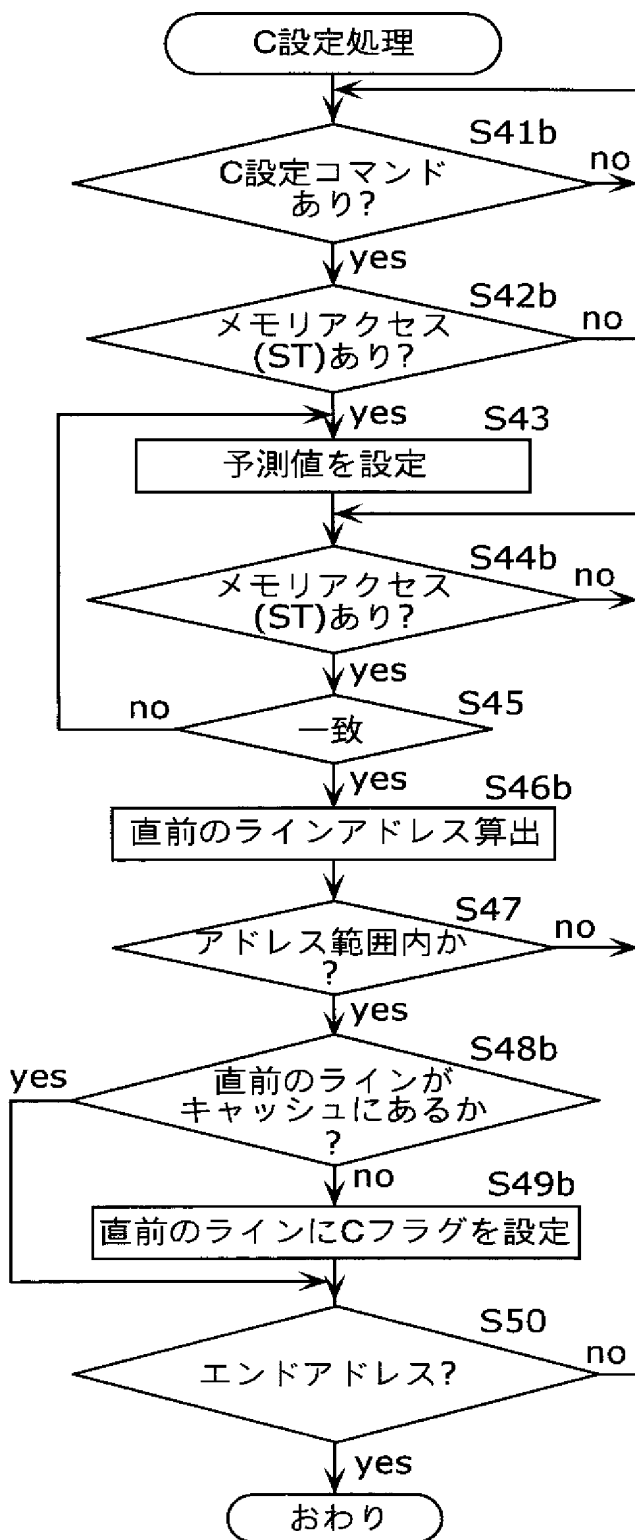


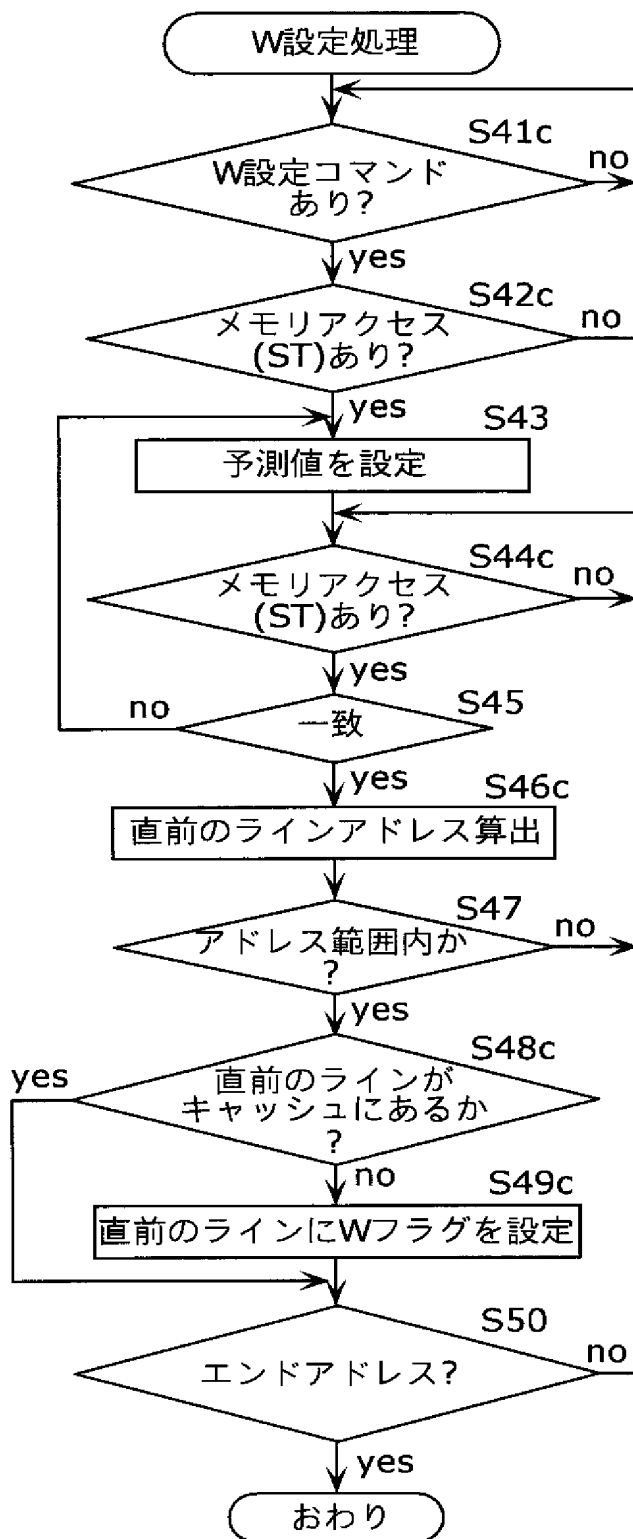
(a)



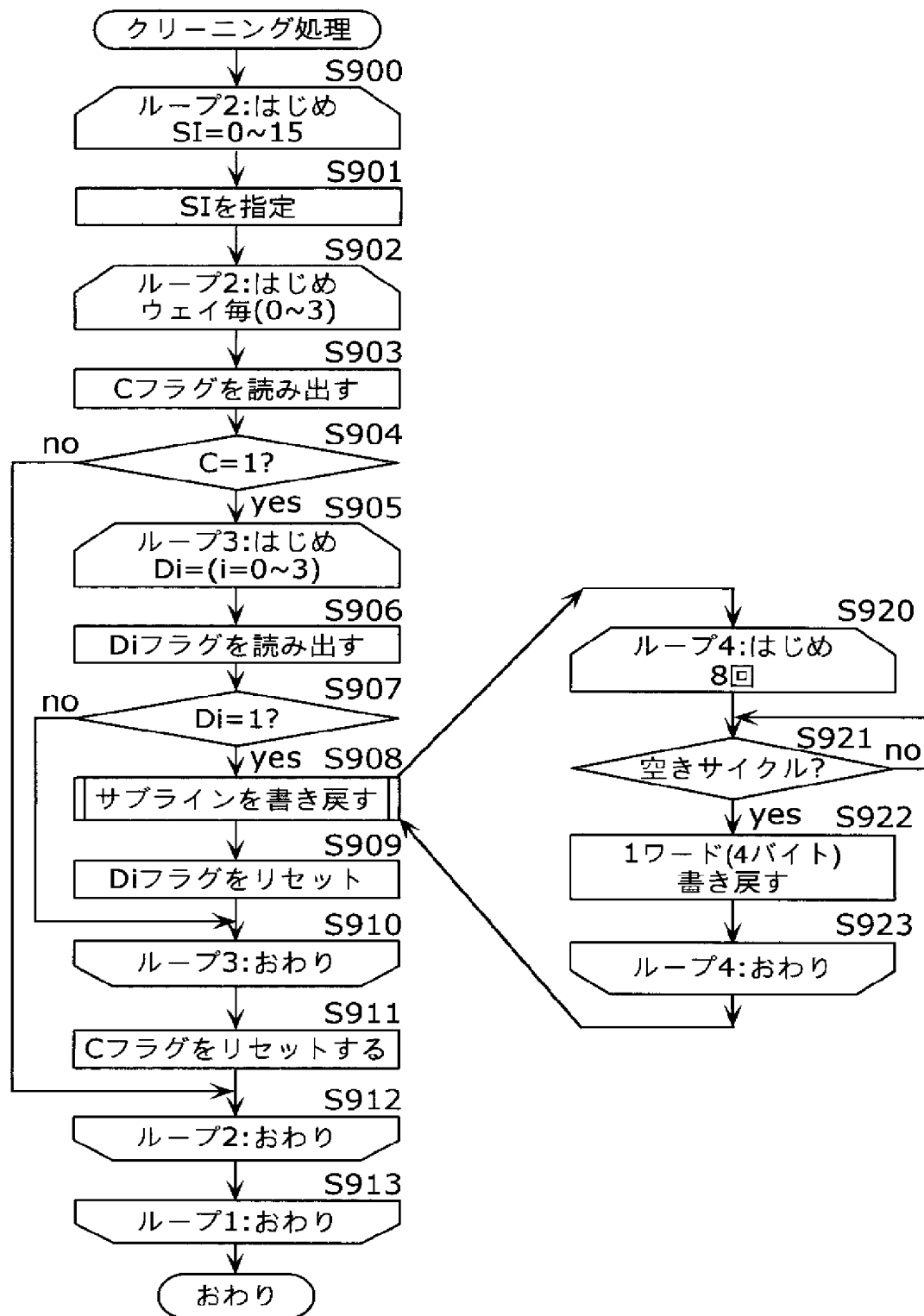
(b)

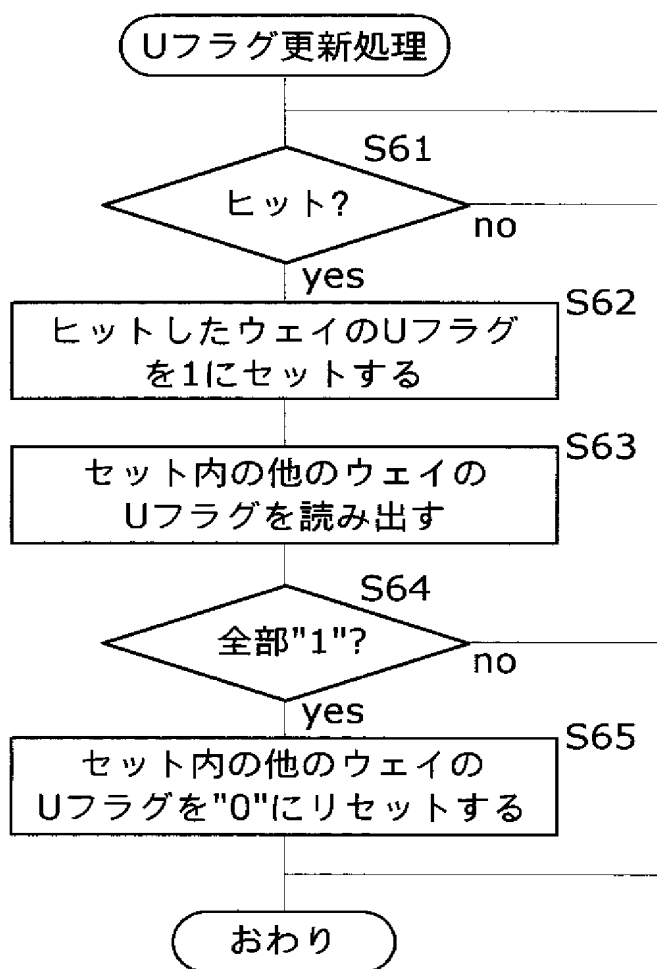


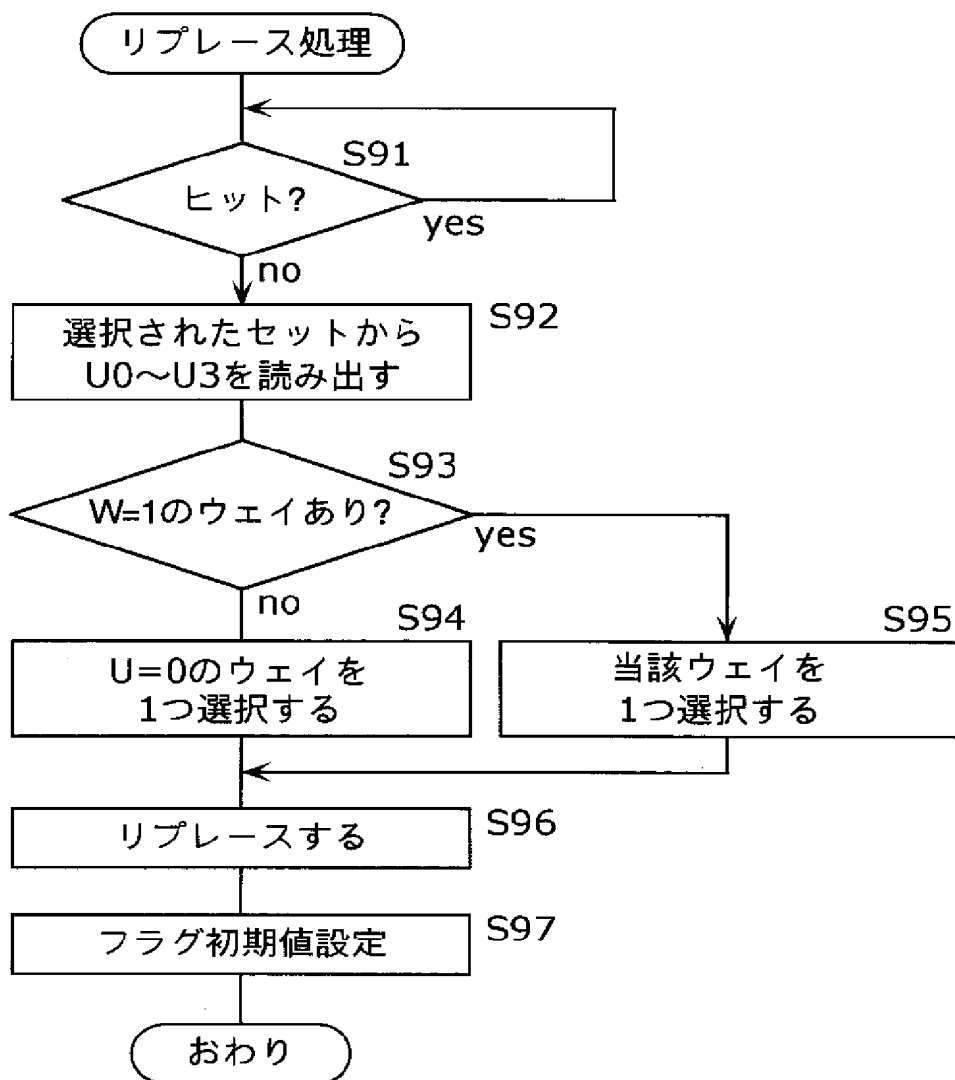




【図 17】







【書類名】 要約書

【要約】

【課題】 ソフトウェア的な性能劣化を招くことなく、プロセッサの動作状況を監視することにより同期を取りつつ適切なタイミングでキャッシュを操作するキャッシュメモリシステムを提供する。

【解決手段】 プロセッサから出力されるメモリアクセスの進行状況に基づいて次にプリフェッチすべきラインアドレスを予測する予測処理部 39 を有し、予測処理部 39 は、メモリからキャッシュメモリに予測されたラインアドレスのデータをプリフェッチするプリフェッチ部 414 と、メモリからキャッシュメモリにデータをロードすることなく、予測されたラインアドレスをタグとしてキャッシュエントリーに設定し、バリッドフラグを有効にするタッチ部 415 とを備える。

【選択図】 図 3

出願人履歴

0 0 0 0 0 5 8 2 1

19900828

新規登録

大阪府門真市大字門真 1 0 0 6 番地

松下電器産業株式会社